1997-03

Development of an on-line failure mode detection and resolution algorithm for the Phoenix AUV

Hurni, Michael A.

Monterey, California. Naval Postgraduate School

NPS-ME-97-001

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

DEVELOPMENT OF AN ON-LINE FAILURE MODE DETECTION AND RESOLUTION ALGORITHM FOR THE PHOENIX AUV

by

Michael A. Hurni

March, 1997

Thesis Advisor: Anthony J. Healey

Approved for public release; distribution is unlimited.

Prepared for:
Office of Naval Research,
800 North Quincy St., Arlington, Virginia 22217-5660

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: DEVELOPMENT OF AN ON-LINE FAILURE MODE DETECTION AND RESOLUTION ALGORITHM FOR THE PHOENIX AUV	5. FUNDING NUMBERS N0001497AF00002
6. AUTHOR(S) MICHAEL A. HURNI	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER NPS-ME-97-001
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research, 800 N. Quincy St., Arlington, VA 22217-5660	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

It has become apparent that in order for an AUV to be a more reliable self-sufficient system, it must have on-line failure detection and resolution capability. In support of this the AUV must have reconfigurable systems so as to be able to take corrective action against resolvable failures. A simulator has been designed using SIMULINK in order to analyze failure modes associated with the NPS Phoenix AUV steering system. The analyses of these failure modes have been used to identify possible signals for steering system fault detection. Finally, a rule based algorithm was developed which can be converted into a format that ultimately could be implemented in a fuzzy logic set, for later insertion into the Phoenix tactical level software. This methodology will be applied to the Navy's UUV.

14. SUBJECT TERMS: AUTONOMOUS SYSTEMS, ROBOTICS, FAULT DETECTION			15. NUMBER OF PAGES 106
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

DEVELOPMENT OF AN ON-LINE FAILURE MODE DETECTION AND RESOLUTION ALGORITHM FOR THE PHOENIX AUV

Michael A. Hurni
Lieutenant, United States Navy
B.S., University of New Hampshire, 1989

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1997

ABSTRACT

It has become apparent that in order for an AUV to be a more reliable self-sufficient system, it must have on-line failure detection and resolution capability. In support of this the AUV must have reconfigurable systems so as to be able to take corrective action against resolvable failures. A simulator has been designed using SIMULINK in order to analyze failure modes associated with the NPS Phoenix AUV steering system. The analyses of these failure modes have been used to identify possible signals for steering system fault detection. Finally, a rule based algorithm was developed which can be converted into a format that ultimately could be implemented in a fuzzy logic set, for later insertion into the Phoenix tactical level software. This methodology will be applied to the Navy's UUV.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PRIMARY NOMENCLATURE

Variable	Description	Units
X,Y	Global position of vehicle	feet
u,v	Velocity of vehicle with respect to a body-fixed reference frame	ft/s
r	Yaw rate of vehicle	rad/s
Ψ	Yaw (heading angle)	radians
Ψ_{com}	Commanded heading angle	radians
$Y_{(.)}$	Hydrodynamic force in body-fixed system with respect to (.)	pounds
$N_{(.)}$	Hydrodynamic moment in body-fixed system with respect to (.)	lbs-ft
m	Mass of vehicle	slugs
x_G	Coordinate of center of gravity in body-fixed system	feet
I_z	Moment of inertia in body-fixed system	ft^4
ρ	Mass density of water	lbs/ft^3
L	Length of vehicle	feet
$\delta_{urb},\delta_{urs},\delta_{lrb},\delta_{lrs}$	Upper and lower bow and stern rudder deflection angles	radians
δ_{com}	Commanded rudder angle	radians
$e_{o(.)}$	Observer error with respect to (.) (measured (.) minus estimated (.))	same as (.)
e_s	Servo error (estimated minus commanded heading angle)	radians
$\dot{(.)}$	Time rate of change of (.)	(.)/s
$\hat{(.)}$	Estimated value of (.)	same as (.)

ACKNOWLEDGEMENTS

It is with the utmost indebtedness that I thank Dr. Anthony Healey, Dr. Fotis Papoulias and Dr. David Marco for their help in working with me on this thesis. Without their vast experience and knowledge, completion of this thesis would have been impossible. I credit most of my research to their high level of technical competence, and to their strong ability to pinpoint and develop an individual's specific talents. I am especially grateful to Dr. Healey for devoting so many hours of his time to my development. He has an uncanny ability to clearly establish realizable goals to achieve a significant productive impact, and he excels in isolating areas in need of improvement. Dr. Healey's ambitious and hard-driving attitude has given me the inspiration needed to charge forward. The result is the possession of the technical and operational knowledge to clearly understand all aspects of marine vehicle dynamics.

I. INTRODUCTION

A. GENERAL BACKGROUND AND LITERATURE

Over the past decade more and more research has been devoted to the development of Autonomous Underwater Vehicles (AUV's). The potential for AUV's is large and eventually, even Remotely Operated Vehicles (ROV's) will incorporate some of the new AUV technology. The potential commercial and military uses for AUV's far out number those of ROV's, not to mention the benefits it can provide to ocean science research. Simply put, AUV's are our future.

Technology of AUV's has come a long way over the last decade. However, much research must still be conducted in all areas of AUV technology. Some of the more recent works are cited here to show the broad spectrum of study that is still ongoing. Healey and Lienard (1993) have shown that a multivariable sliding mode autopilot based on state feedback, designed assuming decoupled modeling, is quite satisfactory for the combined speed, steering, and diving response of a slow speed AUV. Marco and Healey (1996) have demonstrated a method to navigate an autonomous underwater vehicle in a local area using an acoustic sensor for position information derived from feature detection. Further developments have been achieved by Healey, et al. (1994) in hover control behavior using the ST1000 and ST725 high frequency sonars to provide data about the environment. Byrnes, et al. (1996) has proposed a tri-level control system architecture called the Rational Behavior Model (RBM) as an approach to autonomous and automatic control of systems. A work

which describes the advantages of AUV's over ROV's or manned submarines was written by Marco (1996), in which he designed and verified a working hybrid control system combining mission management with robust motion controllers.

An area which is now receiving more focused attention is Failure Modes and Effects Analysis (FMEA). This is because it is becoming more readily apparent that in order for an AUV to be a totally self-sufficient system, it must have an on-line failure detection and resolution algorithm. This algorithm must work in tandem with an AUV whose systems are reconfigurable. The scope of this work takes a very small step towards this goal. Its purpose is to design an effective failure detection and resolution algorithm for one subsystem (steering) of the NPS Phoenix. The Phoenix is an underwater vehicle used by the Naval Postgraduate School to conduct AUV research.

Previous literature pertinent to this particular area of research is very abundant. A brief summary of some basic fault detection methods along with some examples can be found in Isermann (1984). The examples provided were detecting faults in an electrically driven centrifugal pump and leak detection for pipelines. Healey (1993) discusses the use of both batch least squares and Kalman Filters for system parameter identification as a means to detect a change in performance. Bell, et al. (1992) has developed a tool that automates the reasoning portion of an FMEA. The prototype has been created and successfully passed a test and evaluation program. A program which automates the prediction of the effect of failure modes for electrical systems has been developed by Hunt, Price and Lee (1993). This program is called FLAME (Functional

Level Analysis of failure Modes and Effects). Finally, Healey (1992) addresses the use of Kalman Filters and Artificial Neural Networks to provide the detection, and isolation of impending system failures.

For some time the Navy has funded the development of software tools for performing automated on-line system fault detection. It has been agreed, however, that too much detail can lead to unwieldy systems. Also, for a useful automated diagnostic system to be able to improve the reliability of AUV's, only those problems that can be mitigated during a mission need to be identified. Therefore, this work concentrates on "subsystem level" fault detection rather than "component level" detection. The difference lies in the granularity established by the system models used as the detection basis. In principle, a diagnostic system is only able to detect faults at the level of its design model basic granularity. For instance, if a subsystem or component (G) is modeled by its input, output and disturbance signals (Figure 1.1), we can, under some conditions relating to the observability of the system model, infer the satisfactory operation of the subsystem (G) with regard to a desired behavior. If a model based filter is proposed for the control signal (e_o) such that G_c is a model for G, changes in G from any source are detectable through the input/output signals (u, Y) and in particular the servo error ($e_s = Y_{com} - \hat{Y}$) and the observer error ($e_o = Y - \hat{Y}$). The input signal (u) is determined by the system controller (C).

B. SCOPE OF THIS WORK

The driving force of this thesis is threefold: 1) Earlier work by Bahrke (1992) broke down the Phoenix AUV into separate subsystems and generated equations that could be used to model these systems. The work in this thesis seeks to show that these equations can, with reasonable accuracy, model the AUV steering system using a CAD program such as SIMULINK in MATLAB. 2) The simulator designed in this work is used to conduct an intensive study of all the possible failure modes that could be related to the steering system. 3) Armed with the knowledge provided by all the simulations, an algorithm is developed whose purpose is to detect steering system failures and reconfigure the system (when necessary) to operate efficiently, or abort the mission.

Chapter II focuses on the design of an effective, user friendly, FMEA simulator geared towards the NPS Phoenix AUV (Figure 1.2) steering system. It was very important in this chapter to ensure that the vehicle was accurately modeled. However, major emphasis was given toward user friendliness, because future work in this area will require the ability to operate and possibly change the simulator.

Chapter III categorizes the various failure modes that could occur, simulates them and places these results in tabular form to help determine their distinguishing features. Further study is done to show which failure modes can be ignored (small effect on system), which ones must be prevented at all costs (either devastating to the system or undetectable), which ones will require system reconfiguration to ensure

continued effective operation and which ones will require a mission abort. How the steering system should be reconfigured for various failures is also addressed. The ultimate goal of developing a useful algorithm for failure detection and resolution was then realized.

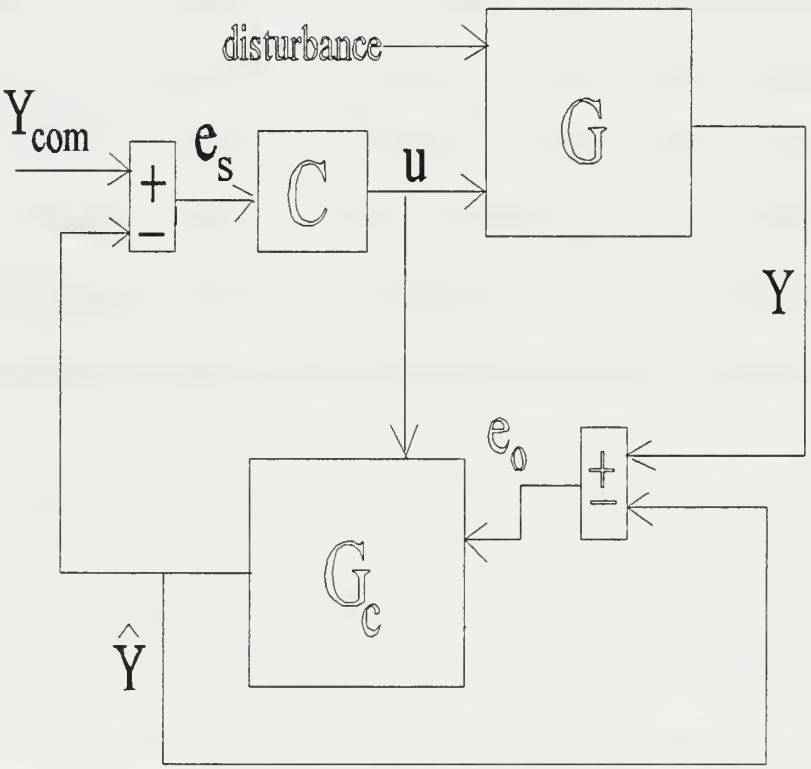

Figure 1.1 Simple Example of Subsystem Modelling.

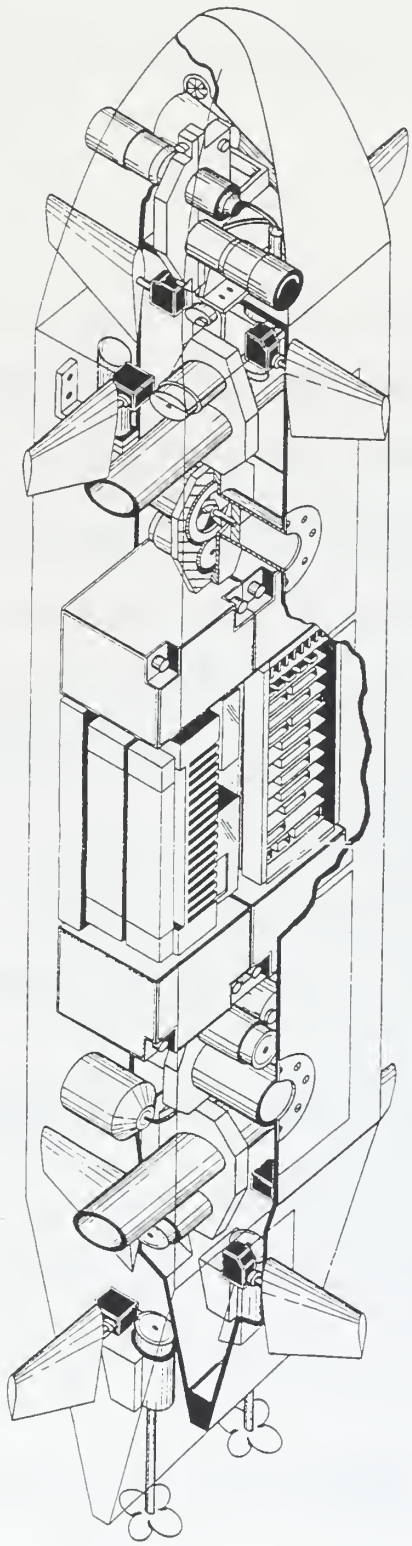

Figure 1.2 Naval Postgraduate School 'Phoenix' AUV. From Marco (1996).

II. DESIGN OF A STEERING SYSTEM FMEA ANALYSIS TOOL

A. INTRODUCTION

The purpose of this chapter is to show the steps taken in the design of a steering system failure modes and effects analysis (FMEA) tool. This FMEA tool is specifically designed to simulate (using MATLAB/SIMULINK) the response of the NPS Phoenix AUV to various failures that could occur in its steering system. Naturally, the design had to be carried out in steps. The dynamics of the vehicle in response to steering system commands had to be simulated first. Once that was achieved, a PD controller was designed, and then a state estimator. Finally, a method of artificially inserting failures into the system was added to the simulation. The overall block diagram is shown in Figure 2.1. Figure 2.2 may be useful in understanding the geometry and axes definitions of the AUV.

B. VEHICLE DYNAMICS RESPONSE TO SPEED AND RUDDER INPUTS

It was shown by Fossen (1994) that the response of the vehicle to speed and rudder inputs can be reduced to a force equation, a moment equation and the three Euler equations relating $\dot{X}$, $\dot{Y}$ and $\dot{\Psi}$. It was assumed that the vehicle operates in level flight without roll, and that body drag forces are negligible. The resulting equations are:

$$m\dot{v} + mx_G\dot{r} - Y_{\dot{r}}\dot{r} - Y_{\dot{v}}\dot{v} = -mur + Y_r ur + Y_v uv + u^2(Y_{\delta_{rb}}\delta_{rb} + Y_{\delta_{rs}}\delta_{rs})$$

$$I_z\dot{r} + mx_G\dot{v} - N_{\dot{r}}\dot{r} - N_{\dot{v}}\dot{v} = -mx_G ur + N_r ur + N_v uv + u^2(N_{\delta_{rb}}\delta_{rb} + N_{\delta_{rs}}\delta_{rs})$$

$$\dot{X} = u\cos\Psi - v\sin\Psi$$

$$\dot{Y} = u\sin\Psi + v\cos\Psi$$

$$\dot{\Psi} = r$$

The following additional assumptions were made by Bahrke (1992):

1. The bow and stern rudders operate with equal magnitudes of deflection but opposite direction ($\delta_{rs} = -\delta_{rb}$).

2. The size and shape of the bow and stern rudders is identical ($Y_\delta = Y_{\delta rb} = Y_{\delta rs}$).

3. From vehicle geometry $N_{\delta rb} = 0.283LY_\delta$ and $N_{\delta rs} = -0.377LY_\delta$.

Along with these assumptions the two rudder inputs (δ_{rs}, δ_{rb}) were divided up into four inputs (δ_{urs}, δ_{lrs}, δ_{urb}, δ_{lrb}) for simulation purposes. This allows any one rudder input to be altered for the purpose of rudder failure insertion and, ultimately, the analysis of the effects of that failure on the vehicle. Using the above relations and rearranging, the force and moment equations can be written as follows:

$$(m - Y_{\dot{v}})\dot{v} + (mx_G - Y_{\dot{r}})\dot{r} = Y_v uv + (Y_r - m)ur + 0.5u^2 Y_\delta(\delta_{urb} + \delta_{lrb} + \delta_{urs} + \delta_{lrs})$$

$$(mx_G - N_{\dot{v}})\dot{v} + (I_z - N_{\dot{r}})\dot{r} =$$

$$N_v uv + (N_r - mx_G)ur + 0.5Lu^2 Y_\delta[0.283(\delta_{urb} + \delta_{lrb}) - 0.377(\delta_{urs} + \delta_{lrs})]$$

In order to model the force and moment equations above, it is necessary to determine the values of the constants and coefficients that appear in the equations.

Appendix A contains the values of these constants and coefficients (Bahrke, 1992). However, the coefficients are in their dimensionless form. Thus it is necessary to dimensionalize the coefficients such that each term in the force equation has units of lbs. and each term in the moment equation has units of lbs.-ft. The coefficients of the Phoenix AUV were dimensionalized by multiplying each one with the appropriate form of $0.5\rho u^{m}L^{n}$ (Healey, 1996). The resulting equations were rearranged such that the force equation was solved for $\dot{v}$ and the moment equation was solved for $\dot{r}$. They are shown here in their final form for simulation:

$$\dot{v} = -0.190597\dot{r} - 0.20901uv - 0.51091ur + 0.011525u^2(\delta_{urb} + \delta_{lrb} + \delta_{urs} + \delta_{lrs})$$
$$\dot{r} = -0.09263\dot{v} - 0.19988ur + 0.0408872u^2[0.283(\delta_{urb} + \delta_{lrb}) - 0.377(\delta_{urs} + \delta_{lrs})]$$

The above two equations along with the equation for $\dot{\Psi}$ were used to form the *Steering* block of Figure 2.1.

The steering system model is shown in Figure 2.3. The inputs to the model are the forward speed of the vehicle (u) and the four rudder deflections ($\delta_{(.)}$). Speed is varied by changing the output value of the *Speed* block of Figure 2.1. The rudders get their commands from the output of the *PD controller* block. The outputs of the model are sway velocity (v), yaw rate (r) and heading angle (Ψ). The last things added to the model were two files containing random, low frequency, system noise (noise1.mat and noise2.mat). These system noise files were added to the right sides of the $\dot{v}$ and

$\dot{r}$ equations. They simulate the added forces and moments that occur naturally due, mainly, to wave action in the water. Appendix B contains the MATLAB program (obtained from professor Healey) used to generate these system noise files.

The *XY output* block of Figure 2.1 was created so that a graph of the vehicle's XY position could be plotted. This graph further aids in understanding the vehicle's response to various failure modes. The internals of this block are shown in Figure 2.4. The XY output is plotted as a function of the inputs u, v and Ψ.

C. PD CONTROLLER DESIGN

In order to design a PD controller the equations for $\dot{v}$, $\dot{r}$ and $\dot{\Psi}$ had to be linearized about a single velocity and then placed in their state space form:

$$\dot{x} = Ax + B\delta_{com}$$

The form that is shown above requires that the four rudder inputs be combined into one commanded rudder input (δ_{com}). The controller design involves pole placement to find a 1x3 matrix (k) such that the control law is δ_{com} = -kx. Linearization was done about u = 1 ft/s and the 3x3 A matrix had one eigenvalue at zero and the other two were complex with real parts at -0.1840. Given these particular eigenvalues the three poles were placed at about -0.2 to approximately match the open loop poles at -0.1840. The resulting gain matrix was k = [0.3587 4.29 0.6949], which produced the following control law:

$$\delta_{com} = -0.3587v - 4.29r - 0.6949\Psi$$

There are two problems with the above control law. The first problem is that it is based on a model that was linearized about u = 1 ft/s. If the vehicle is travelling at u = 3 ft/s the control law is too fast and the vehicle makes a tighter turn than it should. We want a controller that will allow the vehicle to follow the exact same path regardless of its speed. The solution to this problem is to perform the above controller design procedure for various values of u in the hopes of determining a relationship between the controller gains and u. If this relationship can be found then a nonlinear controller design can be obtained with controller gains properly varying as a function of u. The second problem is that v, r and Ψ go to zero in steady state ($\delta_{com} = 0$). It's desirable to have v and r go to zero, but Ψ should reach some commanded heading (Ψ_{com}).

The solution to the first problem is as follows. It was determined that k_1 and k_2 vary inversely proportional to u, however, changing u had absolutely no effect on k_3. The second problem was solved simply by subtracting Ψ_{com} from Ψ. The block diagram of the PD controller is shown in Figure 2.5. Here is the final control law:

$$\delta_{com} = \frac{1}{u}(-0.3587\hat{v} - 4.29\hat{r}) - 0.6949(\hat{\Psi} - \Psi_{com})$$

The inputs to the controller are u, $\hat{v}$, $\hat{r}$, $\hat{\Psi}$ and Ψ_{com}. The *State Estimator* block of

Figure 2.1 provides $\hat{v}$, $\hat{r}$ and $\hat{\Psi}$. The *commanded heading* block of Figure 2.1 provides Ψ_{com}. The controller has two outputs; one for the bow rudders (δ_{com}) and one for the stern rudders ($-\delta_{com}$).

Commanded heading is determined by a switching network (Figure 2.6). It has a *Clock* input which controls the switches to allow the output (Ψ_{com}) to change at the desired time during the simulation. This allows the steering system to receive different heading commands during the simulation.

D. STATE ESTIMATOR DESIGN

The design of the state estimator followed closely the design of the PD controller. The equations for $\dot{v}$, $\dot{r}$ and $\dot{\Psi}$ were linearized about a single velocity and then placed in

the following state space estimator form:

$$\dot{x} = A\hat{x} + B\delta_{com} + Le_{o(.)}$$

It is assumed there are only sensors for r and Ψ, and that v has only an estimated value ($\hat{v}$). Therefore, there are only two observer errors (e_{or} and $e_{o\Psi}$). Since $e_{o(.)}$ is a 2x1 matrix the observer gain matrix (L) must be 3x2. Linearization was done about u = 1 ft/s and pole placement was performed to obtain L. Following normal convention the observer poles were placed at -0.4, which is twice the distance from the origin as

the controller poles. The following observer gain matrix was realized:

$$[L_1, L_2; L_3, L_4; L_5, L_6] = [1.2025, 0.0; 0.4219, 0.0; 1.0, 0.41]$$

Again there is the problem that the state estimator is based on a linear model. If the vehicle is travelling at u = 3 ft/s, the estimator will be too slow. There must be a relationship between u and L. The state estimator design was performed using various values of u, and it was determined that L_2, L_4 and L_5 do not change at all as u is changed. However, it was discovered that L_1, L_3 and L_6 are directly proportional to u. The *State Estimator* block is shown in Figure 2.7. This block has the same inputs as the *steering* block but also has the sensor inputs r and Ψ. It is essentially the same block as the *steering* model with an added term ($Le_{o(.)}$). The outputs are the estimated values of v, r and Ψ ($\hat{v}$, $\hat{r}$ and $\hat{\Psi}$).

E. FAILURE MODE INSERTION

In order to study the effects of various failure modes, a method must be installed to allow failures to be easily inserted into the simulation. Steering system failures can be cataloged into two basic groups: Rudder failures and sensor failures. Rudder and sensor blocks were added to achieve this goal.

1. Rudder Blocks

Four rudder blocks were installed (one for each rudder to match the Phoenix configuration), as can be seen in Figure 2.1. It is relatively easy to modify the

simulation to match other AUV configurations such as that proposed for the Navy's UUV. The diagram of a rudder block is shown in Figure 2.8, and it can be seen that it is a switching network much like the *commanded heading* block. It has two inputs, δ_{com} and *Clock*. The clock input is used to control the switches to insert a failure in δ_{com} at any time during the simulation. It is capable of simulating many failures such as stuck rudder, hard rudder, loose rudder or limited rudder. The output of a rudder block is either the ordered rudder position or the failed rudder position.

2. Yaw Rate Sensor Blocks

Building a block that can accurately simulate the output of a sensor and the output of a failed sensor is again done with the use of switches. The clean output signal is picked off of the *Steering* block and then corrupted with noise. The noise is added to make the simulation more realistic since no sensor can provide a totally noise-free output. The clean signal and the noise each have their own sensor block to allow one of them to be modified during the simulation without affecting the other. For example, a signal with an unusually high noise level could be simulated, or the noise level may be normal but the signal itself is incorrect. Problems that could occur with a sensor besides increased noise are: saturation, loss of input, loss of output and constant bias.

A typical sensor noise block is shown in Figure 2.9. The *Clock* input controls the time at which the problem of high sensor noise would occur (if so desired in the simulation). Basically, the block takes the noise input and adjusts its level for output

to be added to the clean sensor signal. Figure 2.10 shows a sensor signal block. It takes the clean signal, corrupts it and sends the output to be added to the noise signal.

3. Heading Angle Sensor Blocks

Obviously the primary sensor for the steering system is the heading sensor, and it will be proven later that any failure of this sensor is catastrophic. Armed with this knowledge the *Psi sensors* block was added to Figure 2.1. This block assumes that since the heading sensor is so important the vehicle will be equipped with three of them. It is shown in Figure 2.11. It has three sensor and three noise blocks so as to simulate a problem with one or more of the three sensors at a specified time using the *Clock* input. The other two inputs are the clean signal (Ψ) and the estimated signal ($\hat{\Psi}$). Three values of observer error ($e_{o\Psi}$), one for each sensor, are calculated. These observer errors are fed into a function (see Appendix C), similar to a median filter, which extracts any *suspect* errors and averages the rest. A *suspect* observer error is one which exceeds a certain small threshold. The threshold is simply chosen to be slightly greater than acceptable noise amplitudes. This is because during normal steering system operation $e_{o\Psi}$ should only contain zero mean noise. In the case of all three errors being *suspect* (such as when a steering system failure occurs) the function simply averages them all. The output of this block is the average $e_{o\Psi}$ which is sent to the *State Estimator* block of Figure 2.1.

F. SUMMARY

This chapter has shown, step by step, how an FMEA tool was designed so as to study the NPS Phoenix AUV steering system. Much had to be known about the dynamics of the AUV in order to produce a working simulator. The use of switches in sink with a running *Clock* helped make the simulator more user friendly. The user simply has to change the settings of these switches to create the desired simulation. If input files had been used rather than switches, the user would have to edit the files every time the simulation was changed. It has been noted here that there are other ways to insert failures and rudder commands, but the method used in this work was deemed to be the easiest to understand and the most user friendly. The remainder of this work seeks to accomplish a thorough analysis of the Phoenix AUV steering system using the newly designed FMEA tool. The final task will be to establish an algorithm that can be incorporated in the Phoenix software for failure detection and resolution.

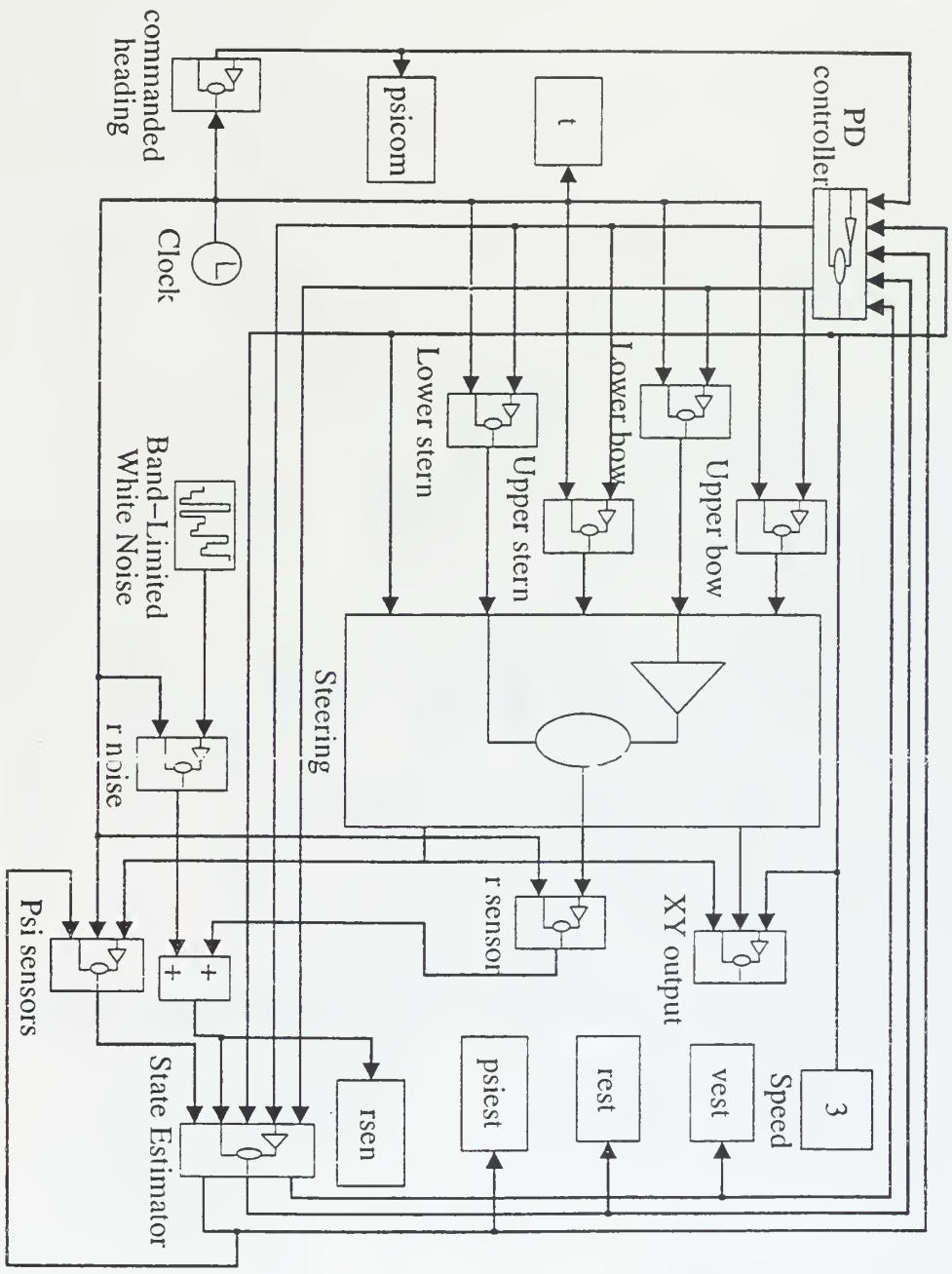

Figure 2.1 Steering System 'Failure Modes and Effects' Analysis Tool.

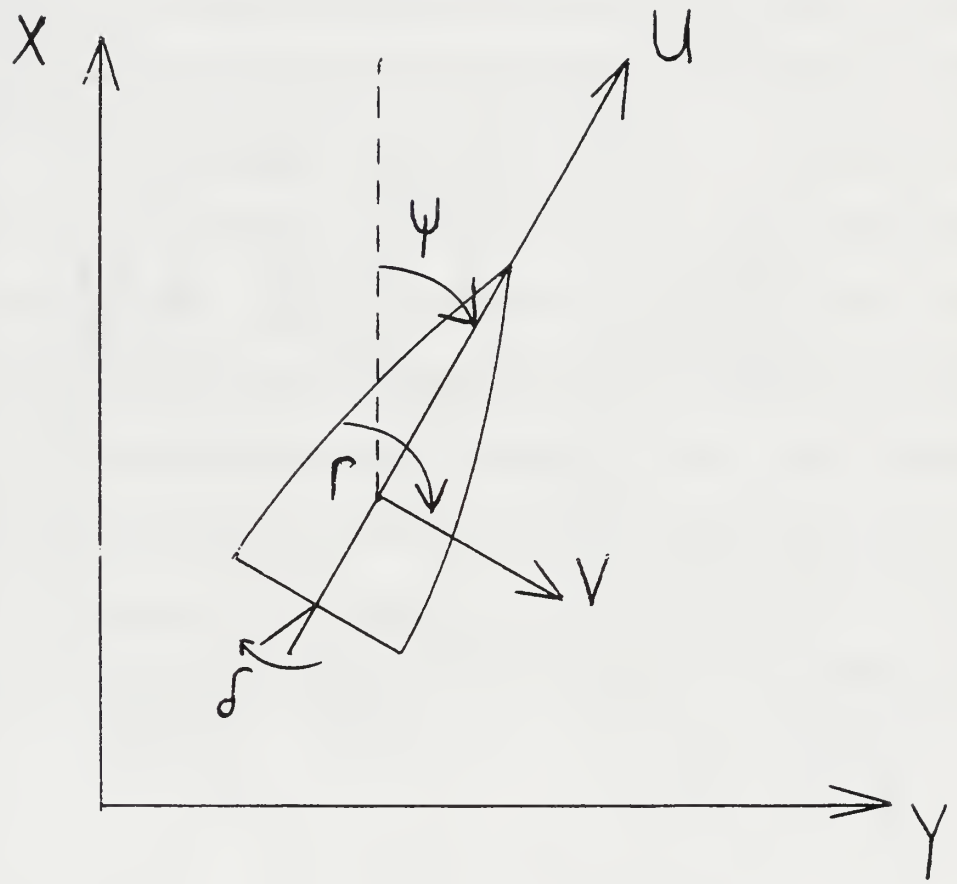

Figure 2.2 Geometry and Axes Definitions (Positive Conventions Shown).

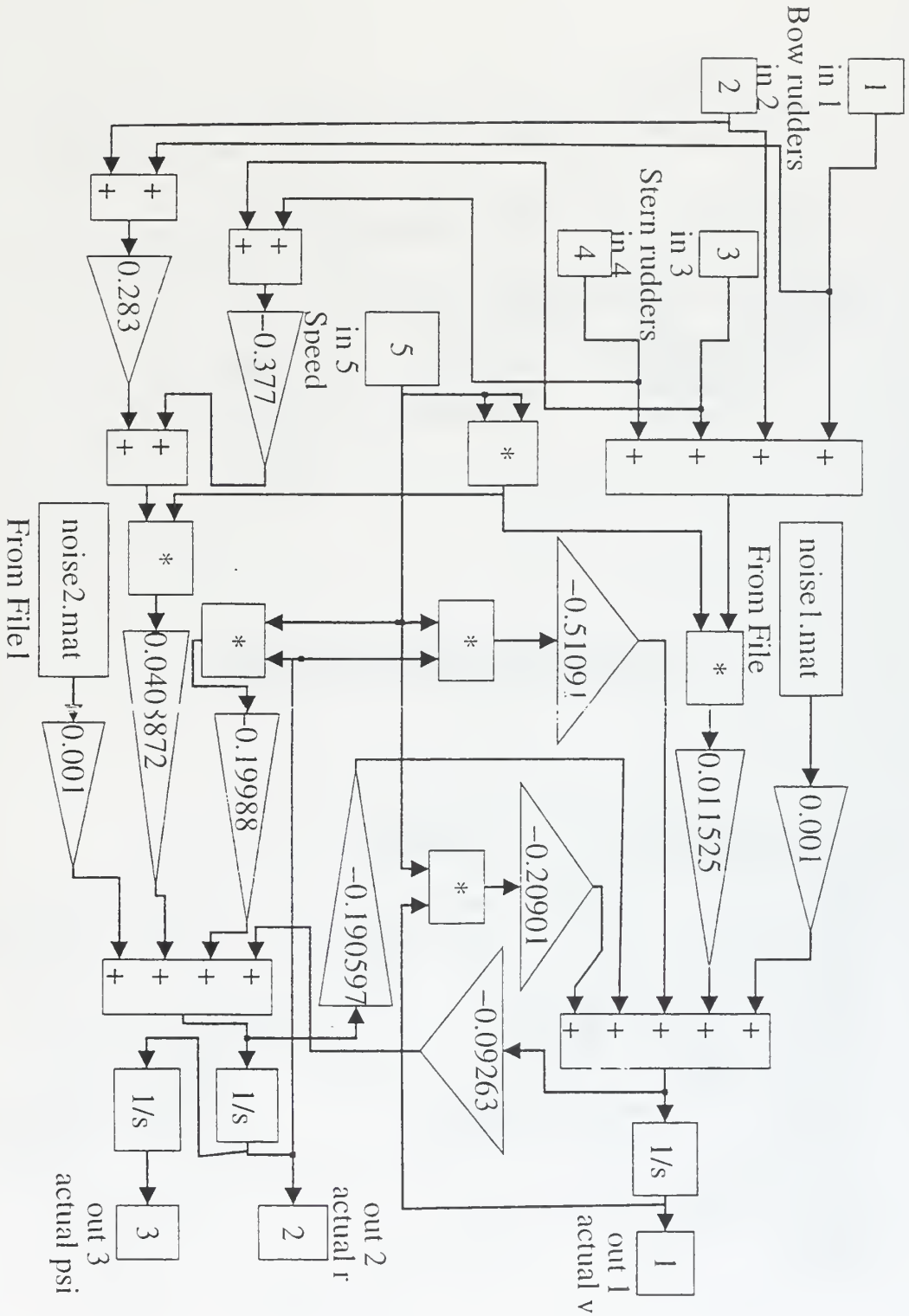

Figure 2.3 Steering System Model.

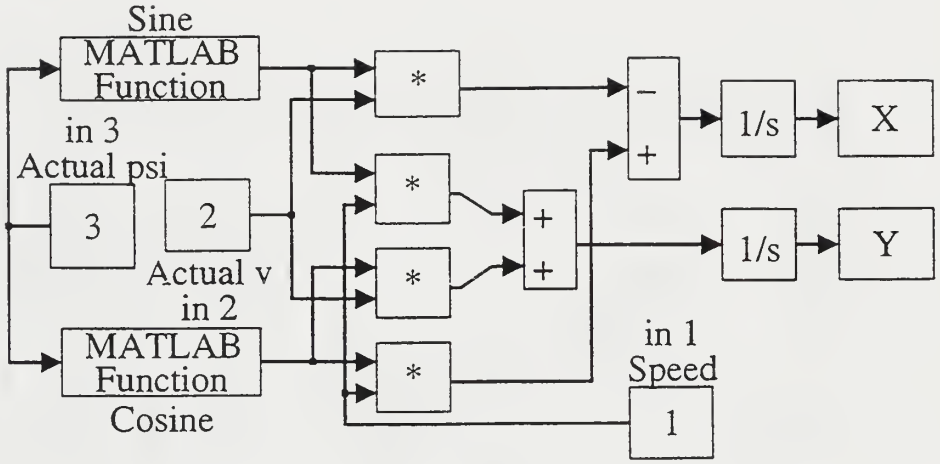

Figure 2.4 Vehicle Position Plotter.

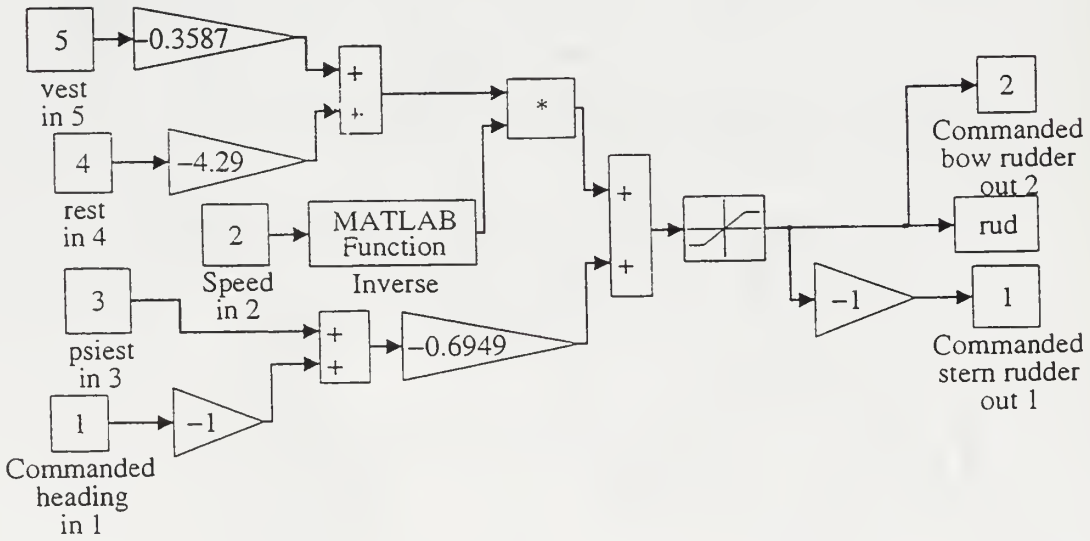

Figure 2.5 PD Controller Block.

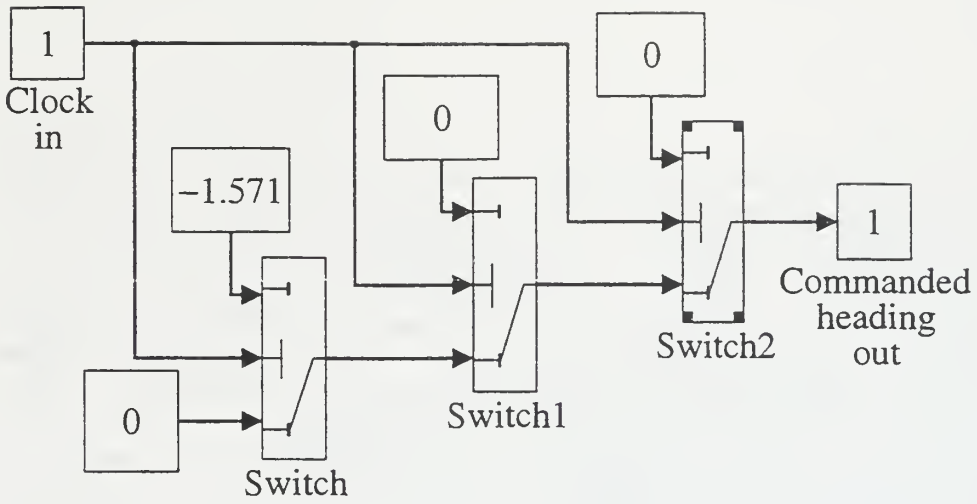

Figure 2.6 Commanded Heading Block.

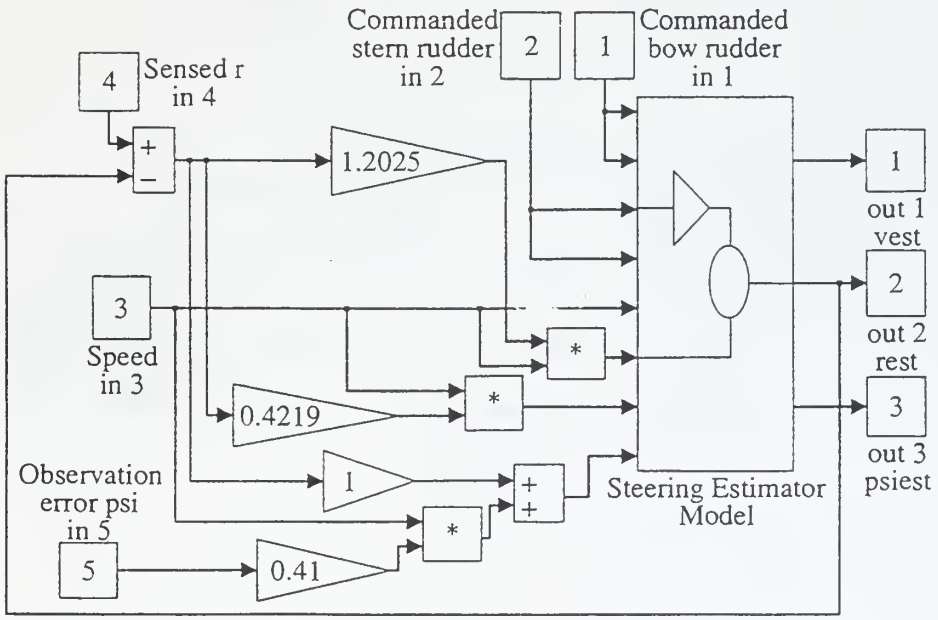

Figure 2.7 State Estimator Block.

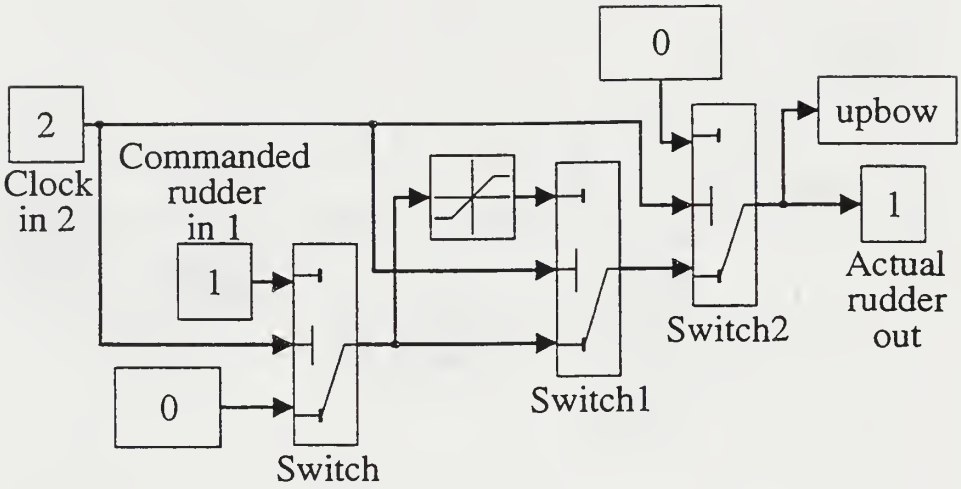

Figure 2.8 Rudder Block.

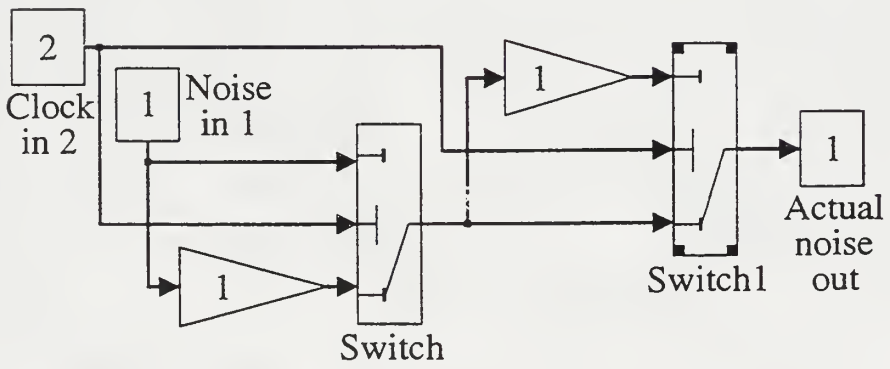

Figure 2.9 Sensor Noise Block.

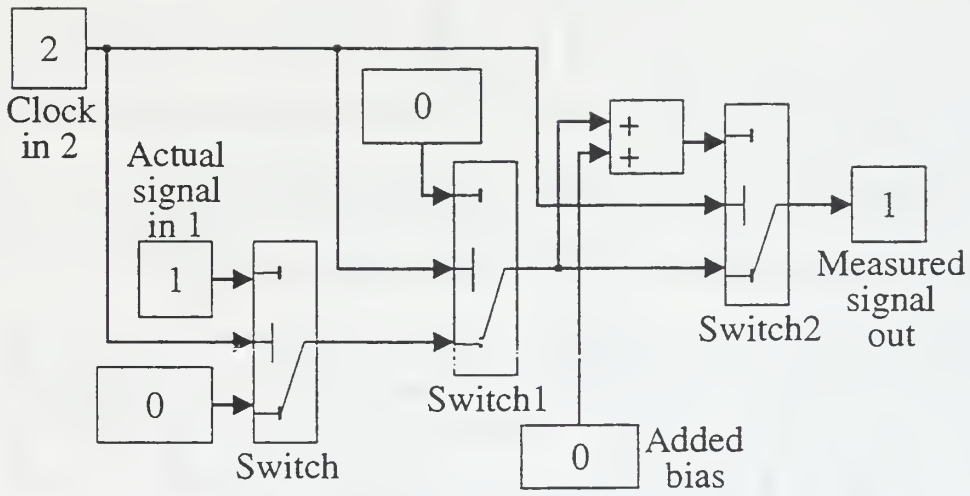

Figure 2.10 Sensor Signal Block.

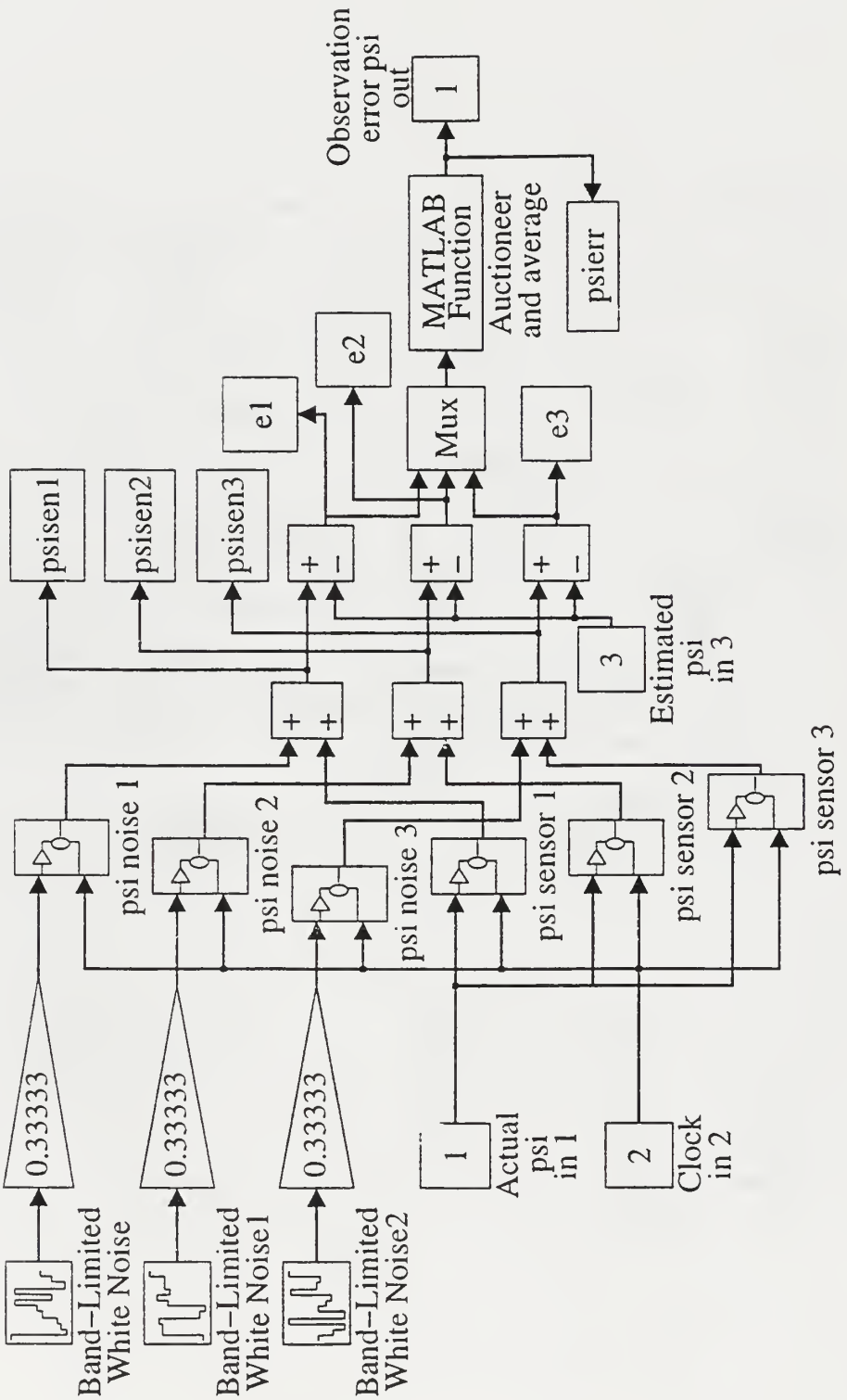

Figure 2.11 Primary Sensor Auctioneering and Averaging Block.

III. PHOENIX AUV STEERING SYSTEM FAILURE MODES ANALYSIS

A. INTRODUCTION

The purpose of this chapter is to show the results of a comprehensive FMEA on the Phoenix AUV steering system. First it was determined what types of failures could occur involving the steering system. Each of these failures was then simulated using the FMEA tool already designed (Chapter II), and each one was compared to the normal (trouble free) response. The results of these simulations were put in tabular form from which they could be studied (for similarities and trends) and further categorized. These studies resulted in many important conclusions involving vehicle dynamics, failure detection and in some cases failure prevention. Further analysis helped formulate possible solutions to the failures. The end result is an algorithm which, when converted to the proper form, can be incorporated in the Phoenix software for failure detection and resolution.

B. FAILURE CATEGORIZATION

In analyzing the AUV steering system it was determined that any failure occurring on the vehicle, if it had any effect on the steering system, could be categorized as one of two types:

1. Rudder Failures
2. Sensor Failures

There are five possible rudder failures that can occur:

(1) Loose Rudder:The rudder in question will not respond (stuck at zero). For the Phoenix (which has four rudders) the other rudders can still turn the vehicle, however its response to course changes will be slower.

(2) Stroke Limited Rudder:The rudder in question does not have its full range of motion. As in the *loose rudder* case, the vehicle will simply respond more slowly to course changes.

(3) Stuck Rudder:A rudder is stuck in some position other than zero; the special case of a rudder stuck at zero is considered in this work as a *loose rudder*. In this case the remaining rudders must turn to counteract the yaw rate caused by the stuck rudder. Once this has been done the vehicle will no longer be heading in the commanded direction, and vehicle response to further course changes will be slowed as in the *loose* or *stroke limited* cases.

(4) Hard Rudder:A rudder is stuck in the *hard right* or *hard left* position. This is simply a special case of *stuck rudder*.

(5) No Rudder Response:All rudders are stuck at zero. In this case the vehicle steering system would not respond at all to commanded course changes.

There are also five possible sensor failures. Since the steering system has two sensors of interest these failures would have to be applied to each sensor, resulting in ten sensor failure modes. The five possible yaw rate sensor failures are as follows:

(1) Increased Noise Level:The more noise that a sensor reads the harder it is to determine the actual value of the parameter being measured. This is a very difficult failure to detect. The intent of modelling this failure is to show that the steering system can operate effectively with as high as a ten fold increase in noise level, thus eliminating the need for detecting it.

(2) Loss of Input:In this case the state estimator will only receive noise from the yaw rate sensor. Since the steering system parameters are fully observable with the heading sensor alone, a loss of the yaw rate sensor should have only a small effect on the vehicle's steering system performance. Detecting this failure may not even be necessary.

(3) Loss of Output:Once the *increased noise level* scenario is proven to have little effect, the results of this failure should be the same as a *loss of input,* since the state estimator is getting zero from the sensor rather than noise.

(4) Sensor Saturation:The state estimator gets a large value of yaw rate. In order to reduce the controller signal to the rudders to zero (so the vehicle doesn't go in circles) the estimated value of heading will be incorrect. The result of this failure should be a large change in heading without the command to do so.

(5) Sensor Bias:The bias in yaw rate will cause an incorrect estimate of heading as in the *saturated sensor* case. Again, the result should be an uncommanded heading change. The amount of heading change will depend on the size of the bias.

It is assumed that since the heading sensor is the primary sensor for the steering system, failures of that sensor should be prevented. However, the following two failure modes were studied in order to show the severity of such failures and the need for prevention:

(1) Stuck Heading Sensor:With a stuck heading sensor the vehicle would respond to course changes by going in circles because it would never reach its commanded heading.

(2) Heading Sensor Bias:With a heading sensor bias the vehicle's course would always be off by the amount of the bias. This failure is impossible to detect. The actual failures studied in this work, along with a complete description of each failure's simulation scenario, are listed in Table 3.1. It should be noted that each simulation starts with the same initial heading and speed (Ψ = 0.0 rads, u = 3 ft/s).

C. PLACING SIMULATIONS IN TABULAR FORM

Each scenario of Table 3.1 was performed twice; once without a failure and once with the failure inserted. The FMEA results are displayed in Table 3.2. The first column

of Table 3.2 lists the failure modes. The next nine columns represent the signals that would be available to the AUV for analysis. The following entries appear in these columns:

'N' ----- No difference between the normal and failed responses.

'S' ----- A steady state difference exists between the normal and failed responses.

'T' ----- A transient difference exists between the normal and failed responses. This means the two responses coalesce in steady state, making detection more difficult.

'I' ----- The difference between the normal and failed responses increases without bound.

Column 11 indicates the failure severity (FS). Entries include:

'L' ----- Low: failure's effect is small; no correction required.

'M' ----- Medium: failure's effect is large enough to require correction such as sensor removal, gain changes and/or bias insertions.

'H' ----- High: failure is not correctable or the AUV can not operate effectively.

The last column in Table 3.2 indicates the difficulty of detection (DD). Possible entries are:

'L' ----- Low: can be detected in steady state using available signals.

'M' ----- Medium: Since the normal and failed responses are equal in steady state, a maneuvering transient is required for detection/verification of failure.

'H' ----- High: not detectable.

D. DISCUSSION OF RESULTS

In order for the AUV to be able to detect failures and respond accordingly, the following questions must be answered. Can all the possible steering system failures be detected? In other words, can we distinguish between a failure and normal operation? It follows that if all failures can not be detected, can we prevent those that are invisible to the AUV's available signals and sensors? Finally, is it possible to distinguish between the detectable failures so that proper action can be taken in response? The answers to these questions will become apparent in the remainder of this work.

In analyzing Table 3.2, it can be assumed that any failed response that has a steady state or transient difference with the normal response can be detected. A difference that increases without bound is considered very easy to detect. Excluding two of the failure modes (high r sensor noise and bias on Ψ sensor) it is possible to detect and distinguish between all steering system failures studied. It will be shown later that high r sensor noise does not have a large enough effect on the vehicle to

even be called a failure. Also, it will be shown that preventing Ψ sensor failures is essential for proper steering system operation.

It may be assumed that when the difference between the failed and normal responses is steady or increasing, there is a failure. The system then has only to determine the exact nature of the failure and what its response will be. But what if the failure only shows up as a transient? It is not good practice to assume a failure has occurred after detecting it once (false alarms do happen). Some failures require a maneuver to verify them, because their normal and failed responses coalesce after the transient is over. Once such a failure has been detected, it is proposed that the maneuver of Figure 3.1 be performed for verification. It is also suggested that this maneuver be executed periodically, whenever the vehicle is not performing any course changes, so as to be able to detect one of these failures early rather than when the vehicle is in a critical situation.

One more thing that can be seen from Table 3.2 is how the observer errors respond to the various failure modes. It should be noted here that e_{or} and $e_{o\Psi}$ are always zero unless a failure has occurred. Figures 3.2 and 3.3 show e_{or} vs time and $e_{o\Psi}$ vs time respectively for the maneuver of Figure 3.1. The only variation in the two signals can be attributed to sensor noise. If Ψ sensor failures are prevented and r sensor noise is not considered a failure, then e_{or} and $e_{o\Psi}$ alone can be used to determine if a failure has occurred. Any non-zero value of e_{or} or $e_{o\Psi}$ can indicate there is a problem. Of course, if this non-zero value was just a transient then the maneuver of Figure 3.1 must be carried out to verify the failure actually exists. Once it has been

determined that there is a problem, e_{or} and $e_{o\Psi}$ can be used together to distinguish whether a rudder problem or an r sensor problem exists. If a rudder problem exists, only e_{or} will be non-zero. If an r sensor problem exists, both e_{or} and $e_{o\Psi}$ will be non-zero.

E. RUDDER FAILURES

1. Loose Rudder

In reality a loose rudder is one that is no longer coupled to its servo mechanism and is at the mercy of the hydrostatic forces in the water. It is assumed here that a loose rudder is one that is stuck at zero. We could even call this a special case of a stuck rudder. The steering system will still operate with a loose rudder, but the vehicle dynamics will be slowed down. In other words, it will take a little longer for the vehicle to perform the desired course change. The severity of this failure mode can be measured by the magnitude of the transient of e_{or}.

If $|e_{or}| < 0.05$, then the ship will still be within one ship length (7.3 ft) of its desired track after the completion of a 90 degree turn, thus no action will be taken (failure severity is low). A 90 degree turn was chosen because this corresponds to the suggested maneuver of Figure 3.1.

If $0.05 \leq |e_{or}| < 0.075$, then a 90 degree turn will result in the vehicle being off its course by more than one ship length but enough rudder response remains to correct the problem. The problem is corrected by increasing the magnitudes of the

controller gains in order to speed up the response of the remaining rudders. This has the effect of speeding up the already slowed dynamics of the vehicle. With the controller gains changed the vehicle can perform its 90 degree turn and still stay within one shiplength of its desired track.

If $|e_{or}| \geq 0.075$, then the failure can not be corrected to keep the vehicle within one ship length after a 90 degree turn. This would be cause to abort the mission.

In the loose rudder scenario studied in this work both bow rudders were stuck at zero. Figures 3.4 and 3.5 show e_{or} = 'T' and $e_{o\Psi}$ ='N'. Also, the magnitude of e_{or} is approximately 0.05. Figure 3.6 shows that the path followed by the vehicle is about one ship length off of the normal path. To verify that the observer errors are correct, the vehicle can check its individual rudder position sensors. Figure 3.7 shows the upper bow rudder to be stuck at zero. This failure can be corrected by increasing the controller gains, thus speeding up the vehicle's turn.

2. Stroke Limited Rudder

For the purposes of this work a stroke limited rudder is one that no longer has its full range of motion. This could be caused by a faulty servo mechanism or a foreign object physically limiting the rudder's travel. Like in the case of a loose rudder, the steering system will still perform its job but vehicle dynamics will again be slowed down. The exact same procedure, as in the loose rudder case, is used to determine the severity of this failure. As before, the correction (if required) will be to

increase the controller gains to speed up the rudder response. However, if $|e_{or}| \geq 0.075$ then there is not enough rudder control remaining to correct the problem and allow the AUV to operate effectively.

In the scenario studied here, both stern rudders are limited to ±0.2 rads. Figures 3.8 and 3.9 show e_{or} = 'T' and $e_{o\Psi}$ ='N'. Since $|e_{or}| < 0.05$ this failure is considered tolerable and a correction is not made. The vehicle is only off course by a few feet as seen in Figure 3.10. Had this failure been more serious and corrective action necessary, it could be verified that the observer errors were correct by checking individual rudder position sensors as in the loose rudder case. Figure 3.11 shows that the upper stern rudder was limited to 0.2 rads rather than saturating at 0.4 rads as it should have if it had its full range of motion.

3. Stuck Rudder

A rudder is stuck if it will not move regardless of the signals sent to the servo mechanism. This can be caused by some foreign object preventing the rudder's movement, or maybe the commanded rudder signal simply is not getting to the servo to change the rudder's position. Since the normal and failed responses of nearly all the signals studied in this work show a definite difference in steady state, no maneuver is required to verify it. However, the actions required are a little more complicated than in the loose or stroke limited cases; and a maneuver will be required anyway to ensure the vehicle can still operate effectively.

The way to combat this failure is to first get the vehicle pointed in the right direction again. This is done by adding a bias to Ψ_{com} that will account for the servo error. This bias is the only one that is added to affect a change in the vehicle's subsystems. Other bias signals must be added for the purpose of removing the steady state differences between the normal and failed responses so that the steering errors are compensated. For example, the controller will still receive $\hat{v}$ from the state estimator, however the failure detection software will get $\hat{v}+v_{bias}$. Other signals besides $\hat{v}$ that require bias additions are $\hat{r}$ (this resets e_{or}) and e_s. Now, as in the loose or stroke limited cases, the steering system dynamics have been slowed because the remaining rudders are compensating for the stuck rudder. So again the maneuver of Figure 3.1 must be executed to see if the controller gains must be raised to allow the vehicle to respond fast enough. In some extreme stuck rudder cases $|e_{or}| \geq 0.075$, at which point the mission must be aborted.

In the stuck rudder scenario from Table 3.1, e_{or} = 'S' and $e_{o\Psi}$ ='N' (Figures 3.12 and 3.13). Individual rudder sensors must be used to verify that the observer errors are not incorrect and to ensure that the wrong failure is not being evaluated. The signals that the AUV would analyze in the case of no rudder response are similar to the stuck rudder case, except for the rudder position. Figure 3.14 shows the upper stern rudder is stuck. The next thing to check is the servo error; if it is increasing without bound then the vehicle can not correct the stuck rudder problem using the

other rudders and the mission would be aborted. In this case, however, servo error does not increase without bound (Figure 3.15). Now the vehicle must be pointed in the right direction and the proper biases must be inserted. Then the maneuver of Figure 3.1 must be performed to determine if controller gain change is required. In this case $|e_{or}| < 0.05$ (see Figure 3.12), so no gain changes are necessary. The vehicle's path for this scenario is shown in Figure 3.16.

4. Hard Rudder

Simply put, a hard rudder is just a special case of a stuck rudder. In this case the rudder is stuck at an angle of ±0.4 rads (the maximum rudder angle allowed). It can be caused by the same things that created the stuck rudder. It can also be caused by a wire that has shorted itself to some source of voltage that would then saturate the signal going to the servo mechanism. Since this is still a stuck rudder case the indications are the same; and the actions required to combat the failure are identical to those of the stuck rudder case.

In the scenario studied here (Table 3.1) both stern rudders are stuck hard at 0.4 rads. Again it is shown that e_{or} = 'S' and $e_{o\Psi}$ ='N' (Figures 3.17 and 3.18), as in the stuck rudder case. Rudder sensors are checked to ensure the observer errors are correct and that it is not a 'no rudder response' case. The servo error is then checked (Figure 3.19) and it is found to be increasing without bound. This makes sense because the stern rudders have a larger effect on vehicle steering than the bow rudders. This is due to the larger moment arm of the stern rudders, since they are located

further from the center of gravity than the bow rudders. The mission must be aborted since the vehicle can only go in circles (Figure 3.20).

5. No Rudder Response

In this case none of the rudders move in response to a commanded course change. This could happen if all four rudders were simultaneously stuck at zero or if $\delta_{com} = 0$ due to a wire being shorted to ground. In the scenario of Table 3.1 the vehicle first detects that e_{or} = 'S' and $e_{o\Psi}$ ='N' (Figures 3.21 and 3.22). The rudder sensors are then checked and it is found that they are all stuck at zero. Figure 3.23 shows the upper stern rudder response and it is in fact stuck at zero. Also, the vehicle continues to move in the same direction despite the commanded course change (Figure 3.24). The wiggle of Figure 3.1 can be attempted to see if maybe this is a temporary condition, but if no response is noted by the rudders the mission must be aborted.

F. YAW RATE SENSOR FAILURES

1. Increased Yaw Rate Sensor Noise

Many different types of interference can increase sensor noise. The worst kind of interference can come from jamming, which is very possible if the AUV is used for military purposes such as mine countermeasures. In this work a 90 degree heading change was ordered while the r sensor experienced a noise increase by a factor of ten. From Figure 3.25 it can be seen that the vehicle's path was nearly unaffected by the increased noise. It is therefore assumed that a noise increase in the r sensor does not

constitute a failure and the detection software will not even consider it an option when conducting failure analysis.

2. Loss of Yaw Rate Sensor

If the vehicle experiences a loss of r sensor input, the state estimator will simply receive sensor noise. Similarly, if the entire sensor is lost (no output) the state estimator will receive an input of r = 0. Both cases are lumped into one here since the sensor noise has little effect anyway. In the scenario found on Table 3.1, the loss of input to the r sensor is used to study this failure. Figure 3.26 shows that by losing the r sensor the vehicle can still operate effectively. Given a 90 degree turn it is off course by only about six feet (less than a ship length). Therefore, this is not a failure that even needs to be accounted for by the detection software. However, we now know that if a more severe failure occurs in the r sensor, the solution will be to ignore the sensor's output and call it zero.

3. Saturated Yaw Rate Sensor

A saturated r sensor could occur when there's a short in the sensor circuit. The severity of this failure on the vehicle's path can be seen in Figure 3.27. To determine this failure has occurred the vehicle circuitry would detect that e_{or} = 'S' and $e_{o\Psi}$ ='S' (Figures 3.28 and 3.29). It can be verified that the observer errors are accurate by checking that there is a steady state e_s (Figure 3.30) while δ_{com} steadies out to zero (Figure 3.31). The response to this failure is to ignore the r sensor output and set it to

zero. This is the chosen action since it has already been shown that the vehicle can operate effectively without the sensor.

4. Bias On Yaw Rate Sensor

The effect on the vehicle's path of a bias on the r sensor is shown in Figure 3.32. Again e_{or} = 'S' and $e_{o\Psi}$ ='S' (Figures 3.33 and 3.34). As in the saturated sensor case, verification can be obtained by showing that there is a steady state e_s (Figure 3.35) while δ_{com} steadies out to zero (Figure 3.36). The proper response would be to operate without the sensor, as was done in the saturated case.

G. HEADING ANGLE SENSOR FAILURES

The study of Ψ sensor failures is simple and straight forward. The Ψ sensor is the primary sensor of the steering system, so its importance can not be overemphasized. Since it is desired that failures not occur with this sensor, it is proposed that the vehicle be supplied with three of them. This does not mean three separate inertial units are required; magnetic compasses are low cost and reasonably accurate. When one sensor fails there will still be two more to supply the correct heading. The two failure modes studied in this work were chosen to show the importance of having three sensors. With this in mind, there will be no discussion as to how it will be detected or what actions will be taken. With three sensors on board it will be assumed that failures of the Ψ sensor will not occur.

With a stuck Ψ sensor the vehicle can not measure its heading (knowing only one direction). As soon as a Ψ_{com} other than the stuck position of Ψ is ordered the vehicle will simply move around in circles (Figure 3.37). This would be difficult to detect and correct since e_s (Figure 3.38) and r (Figure 3.39) contradict each other. The constant e_s would indicate the vehicle is traveling in a straight line, but the constant r indicates it is moving in a circle.

An unknown sensor bias added to Ψ would cause the vehicle to move in the wrong direction. It would alter the AUV's course by an amount equal to Ψ_{bias} (Figure 3.40). What is even worse than this is the fact that this failure mode is completely undetectable. Figures 3.41 through 3.44 show e_{or}, $e_{o\Psi}$, e_s and δ_{com}, respectively. The only significant difference between the normal and failed responses occurs at five seconds when the bias is inserted. Once the bias is inserted, the normal and failed responses are identical. If the bias already existed, it would not be detected. Similarly, if the occurrence of the bias was caught it would not be able to verify it. It is possible that a problem like this could be noted on successive navigational fixes, but then the system would have to be able to distinguish between Ψ_{bias} and normal set and drift.

H. SUMMARY

It must be noted here that when the AUV controller detects a failure and responds accordingly, it does not necessarily know the exact nature of the failure. For example, whether the r sensor is saturated or has a bias is not a concern. The system

need only know that there is an r sensor problem so that its value can be ignored and set to zero for the estimator. Similarly, the procedure used to correct a rudder failure seeks only to ensure the vehicle can operate effectively. It does not care which rudder is actually causing the problem. This extra information can be determined when the mission is over and the vehicle can be tested.

In this chapter a comprehensive FMEA was conducted on the Phoenix AUV steering system. Several possible failure modes were analyzed. The results of these analyses provide methods to detect the various failures and what to do when they occur. These results have been ordered into a useful algorithm to be incorporated in the AUV's software for failure detection and resolution. This algorithm has been included in this work as Appendix D.

Remember that this chapter deals only with the steering system. The algorithm of Appendix D would obviously be running in conjunction with other failure detection algorithms. For example, there should also be algorithms for the propulsion and diving systems. It is a fact that when there is a stuck rudder (and the other rudders are turned to make up for this) the drag from the rudders will slow the vehicle down. Also, the act of conducting a turn will slow down the vehicle. Speed was not considered to be a signal monitored by the steering system. However, the propulsion system could be used to distinguish between a stuck rudder and no rudder response, rather than measuring all the individual rudder position sensors. A reduction in speed, as sensed by the propulsion system, could indicate the rudders have moved. Finally,

in the case of a rudder failure (such as a stuck rudder), the propulsion system may also need to take action, such as increasing propeller rpm's to maintain speed.

The last issue needing some attention is the action taken when the failure detection system is not working properly. What happens if the detection system reads a combination of signals that it can not map to a specific failure mode? The worst case must always be assumed. In this case it must be assumed that the failure detection system is not working properly. Therefore, if a failure occurs it may not see it. There can only be one solution to this dilemma: abort mission.

Table 3.1 Failure Scenarios

Failure Mode	Failure Scenario
Loose rudder	Ψ_{com} changed to -1.571 rads at 5 seconds. δ_{urb} and δ_{lrb} stuck at zero
Stroke limited rudder	Ψ_{com} changed to -1.571 rads at 5 seconds. δ_{urs} and δ_{lrs} limited to $\pm$ 0.2 rads
Stuck rudder	Ψ_{com} changed to -1.571 rads at 30 seconds. δ_{urs} stuck at 0.2 rads at 5 seconds
Hard rudder	δ_{urs} and δ_{lrs} go hard over to 0.4 rads at 5 seconds
No rudder response	Ψ_{com} changed to -1.571 rads at 5 seconds. All rudders stuck at 0.0 rads
Increased r sensor noise	Ψ_{com} changed to -1.571 rads at 30 seconds. Sensor noise increased 10X throughout simulation
Loss of input to r sensor	Ψ_{com} changed to -1.571 rads at 5 seconds. No sensor input (reading noise only)
Saturated r sensor	Sensor saturates (r = 0.873 rad/s) at 5 seconds
Bias on r sensor	Ψ_{com} changed to -1.571 rads at 20 seconds. Bias of 0.2 rad/s inserted at 5 seconds
Stuck Ψ sensor	Ψ_{com} changed to -0.524 rads at 5 seconds. Sensor stuck at 0.0 rads
Bias on Ψ sensor	Ψ_{com} changed to -1.571 rads at 30 seconds. Bias of 0.2 rads inserted at 5 seconds

TABLE 3.2 FMEA RESULTS

Failure Mode	e_{or}	$e_{o\Psi}$	e_s	δ_{com}	$\hat{r}$	r	$\hat{\Psi}$	Ψ	$\hat{v}$	FS	DD
Loose rudder	T	N	T	T	T	T	T	T	T	M	M
Stroke limited rudder	T	N	T	T	T	T	T	T	T	L	M
Stuck rudder	S	N	S	S	S	T	S	S	S	M	L
Hard rudder	S	N	I	S	S	S	I	I	S	H	L
No rudder response	S	N	S	S	S	T	S	S	S	H	L
Increased r sensor noise	N	N	N	N	N	N	N	N	N	L	H
Loss of input to r sensor	T	T	T	T	T	T	T	T	T	L	M
Saturated r sensor	S	S	S	T	S	S	S	S	S	M	L
Bias on r sensor	S	S	S	T	S	S	S	S	S	M	L
Stuck Ψ sensor	N	S	S	S	S	S	S	S	S	H	L
Bias on Ψ sensor	N	N	N	N	N	N	N	N	N	H	H

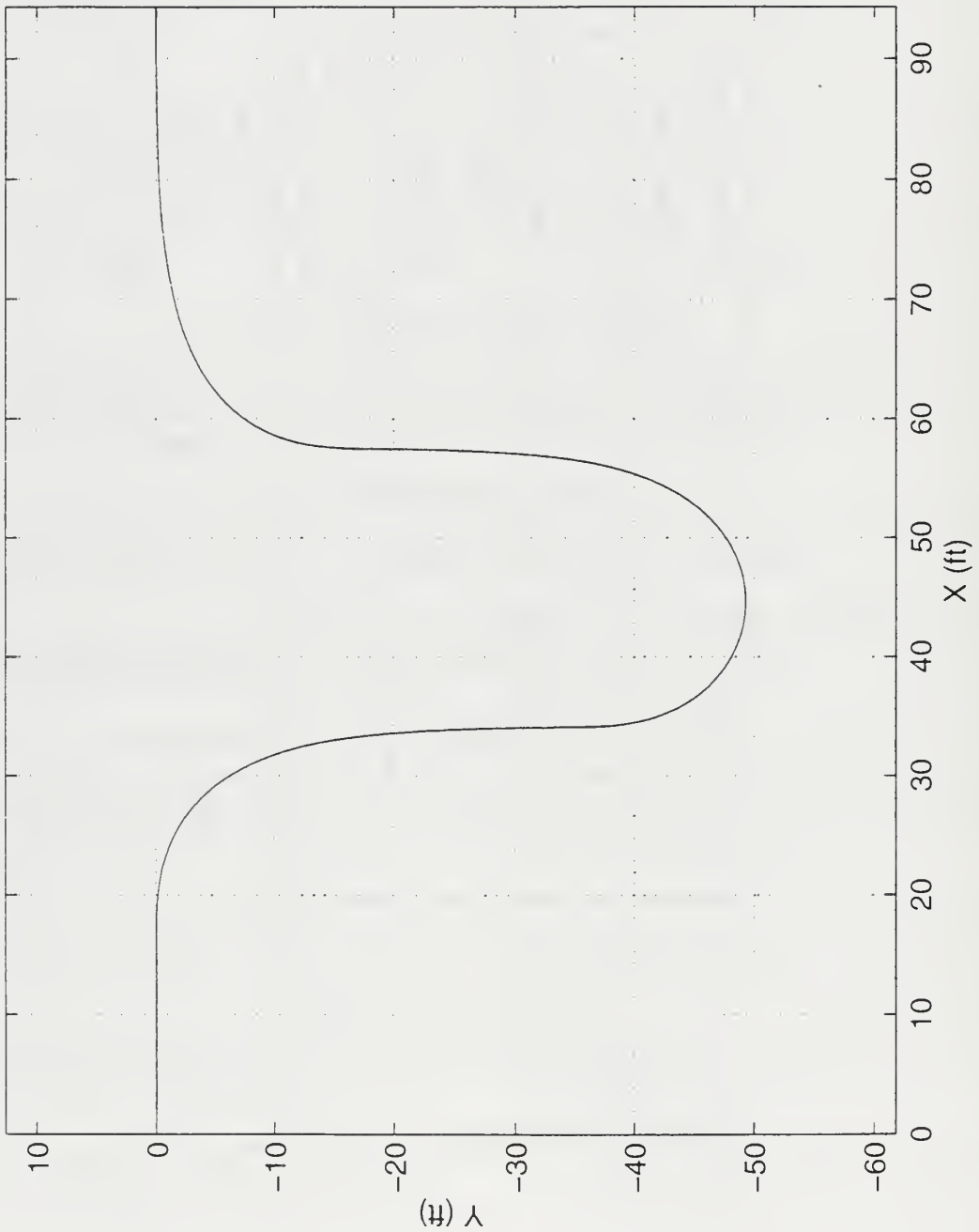

Figure 3.1 Vehicle Position Plot for a Typical Maneuver

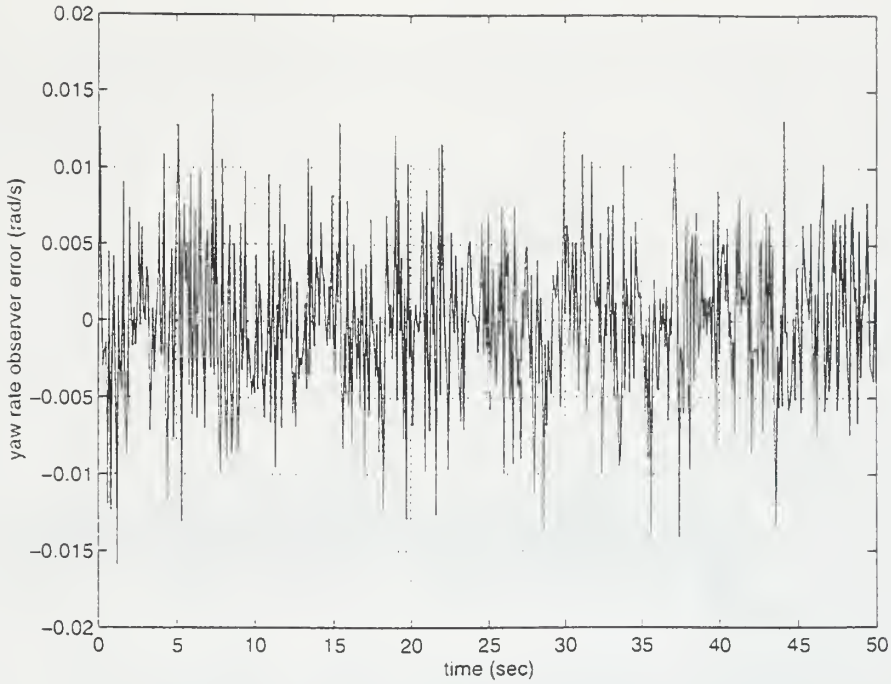

Figure 3.2 Typical Maneuver (e_{or} vs time).

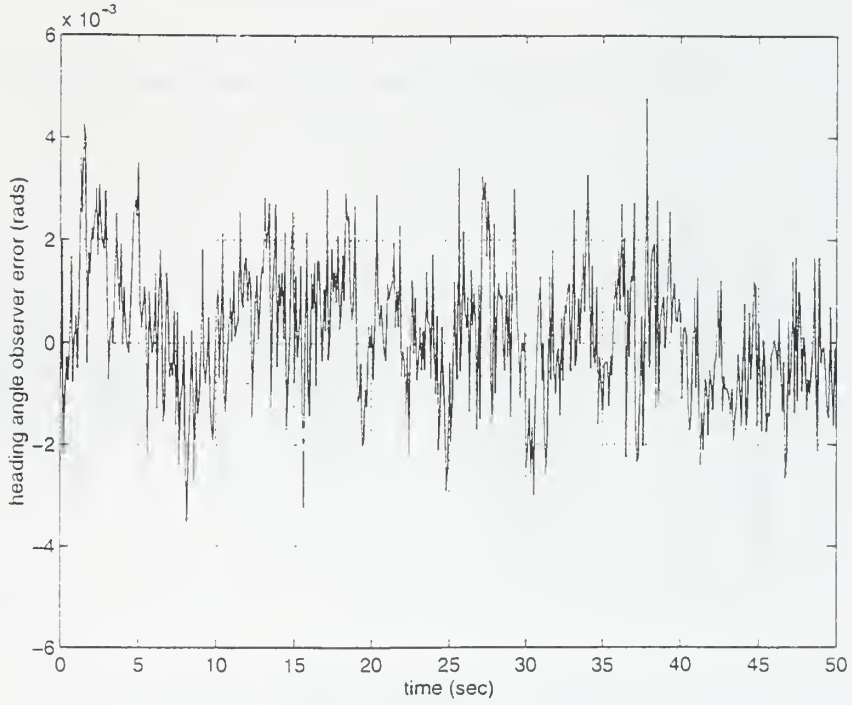

Figure 3.3 Typical Maneuver ($e_{o\Psi}$ vs time).

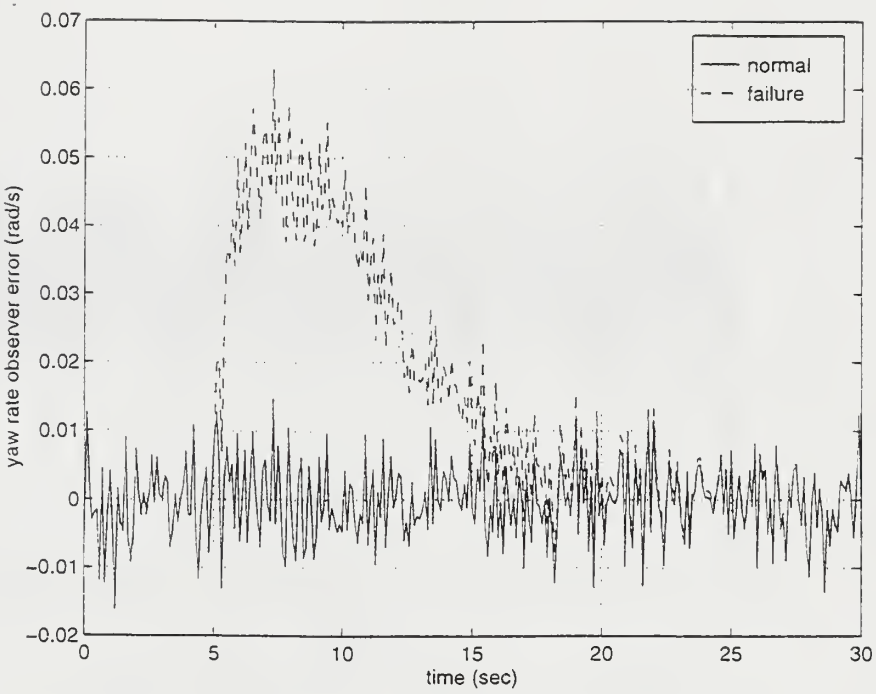

Figure 3.4 Loose Rudder (e_{or} vs time).

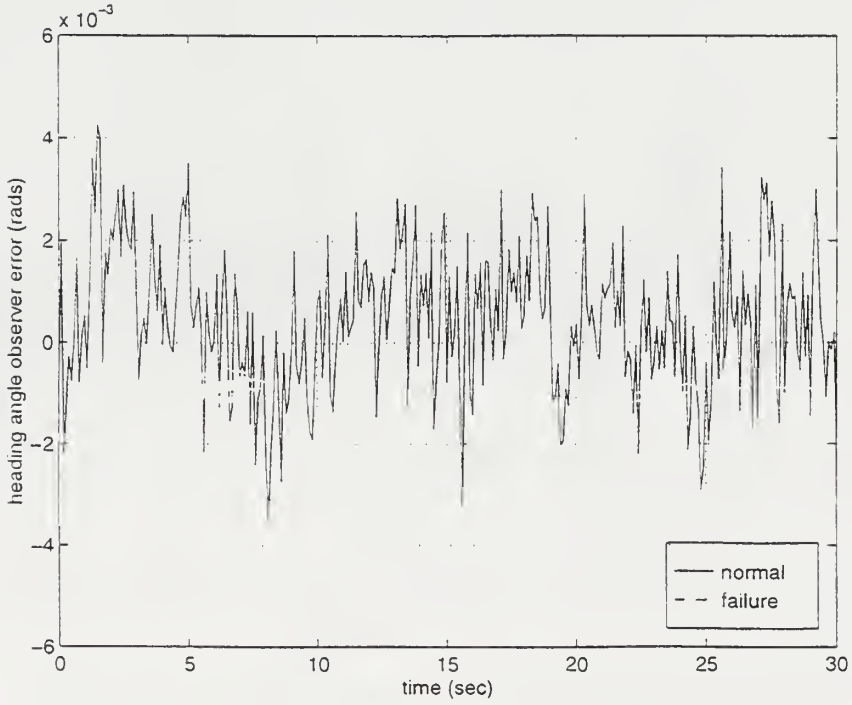

Figure 3.5 Loose Rudder ($e_{o\Psi}$ vs time).

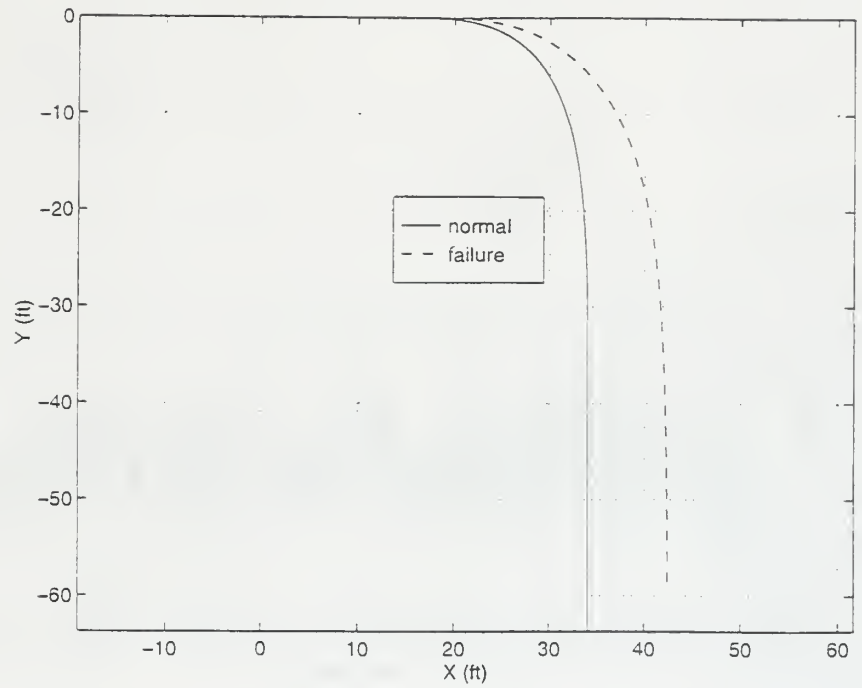

Figure 3.6 Loose Rudder (Position Plot).

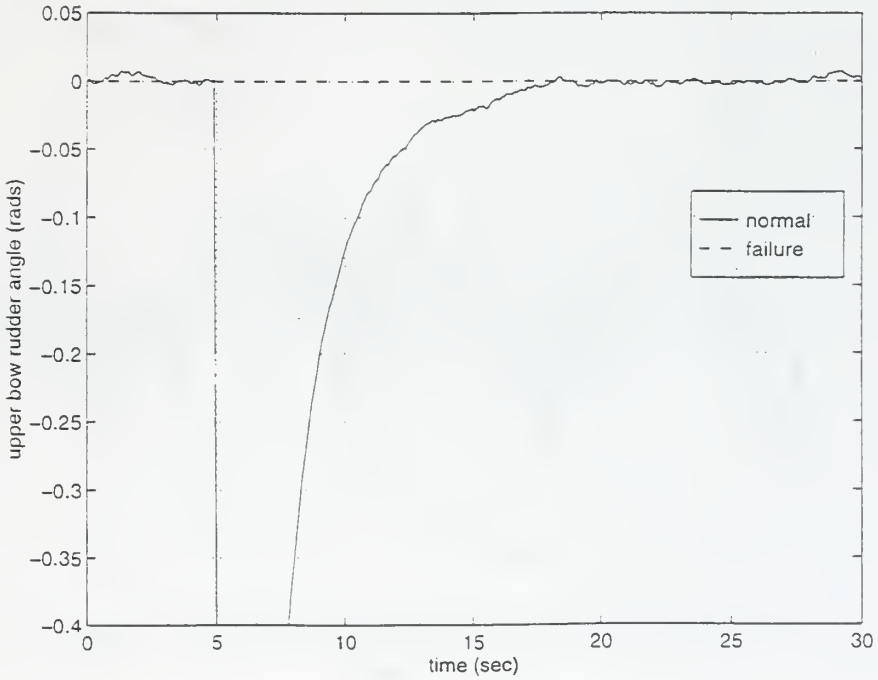

Figure 3.7 Loose Rudder (δ_{urb} vs time).

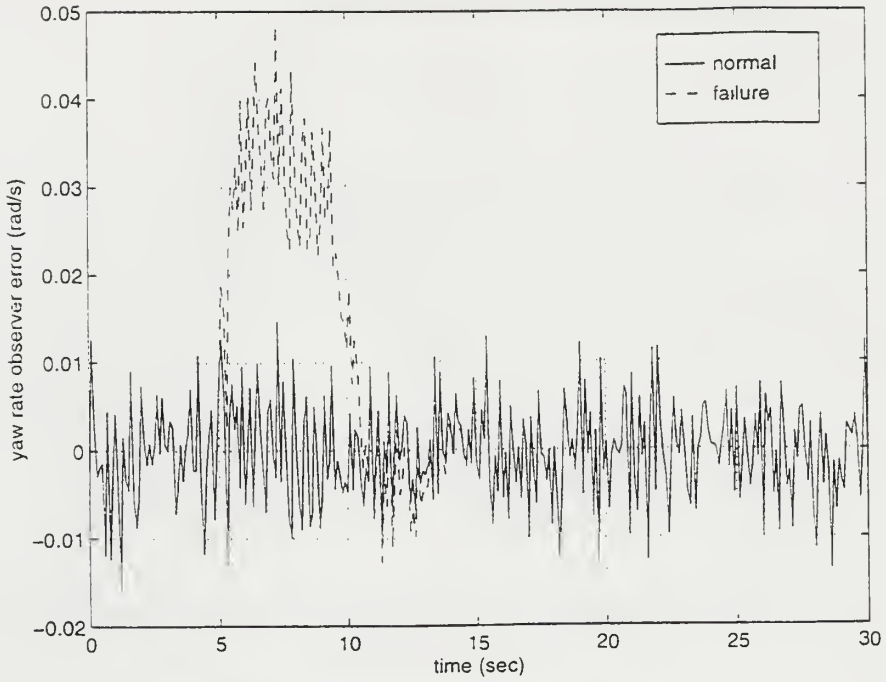

Figure 3.8 Stroke Limited Rudder (e_{or} vs time).

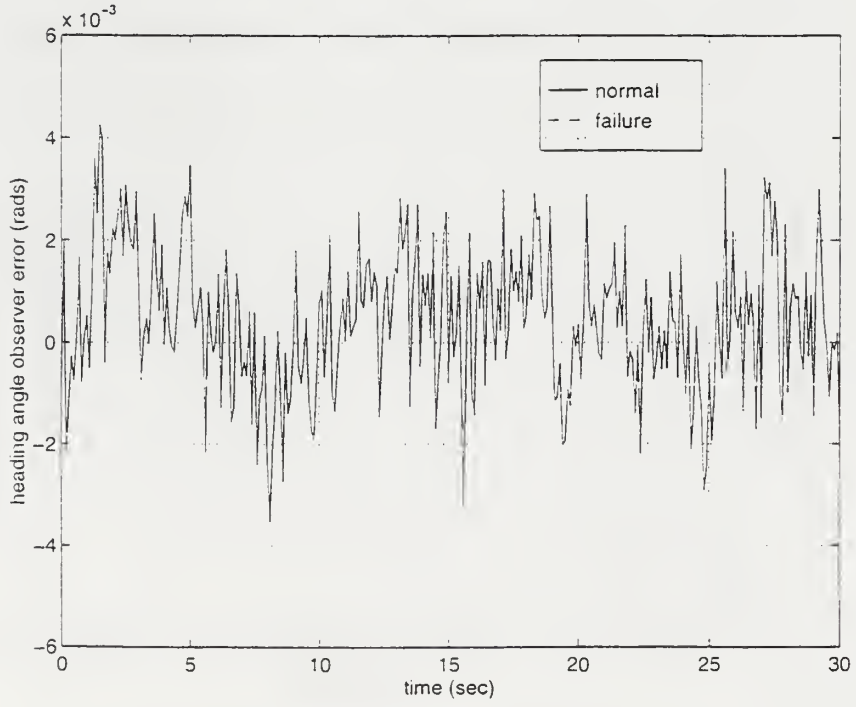

Figure 3.9 Stroke Limited Rudder ($e_{o\Psi}$ vs time).

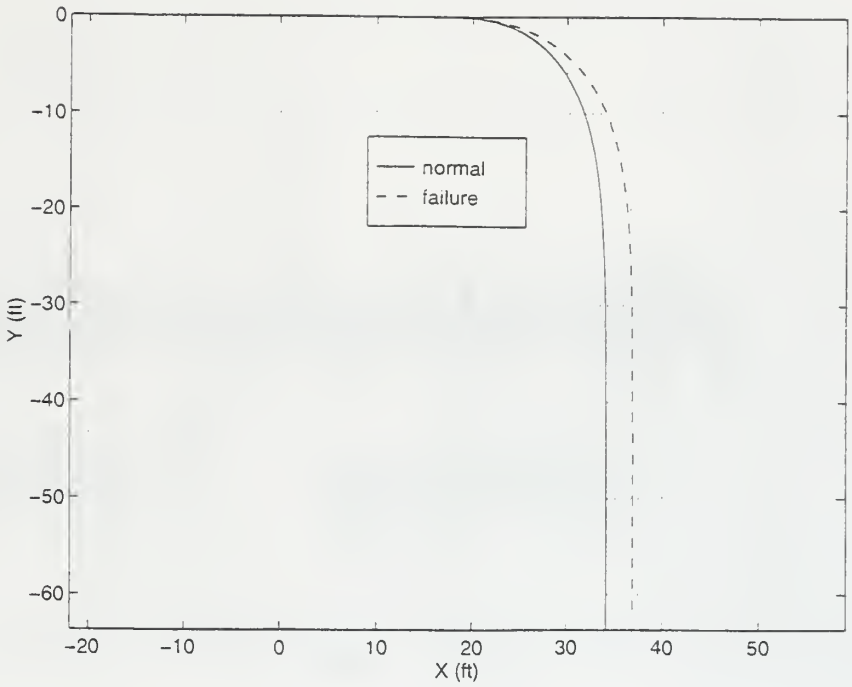

Figure 3.10 Stroke Limited Rudder (Position Plot).

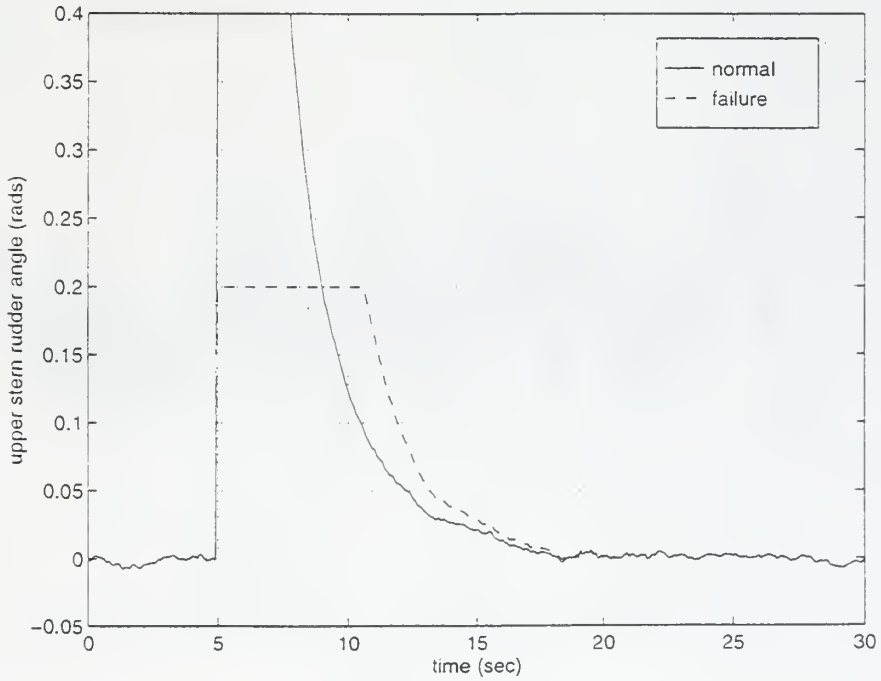

Figure 3.11 Stroke Limited Rudder (δ_{urs} vs time).

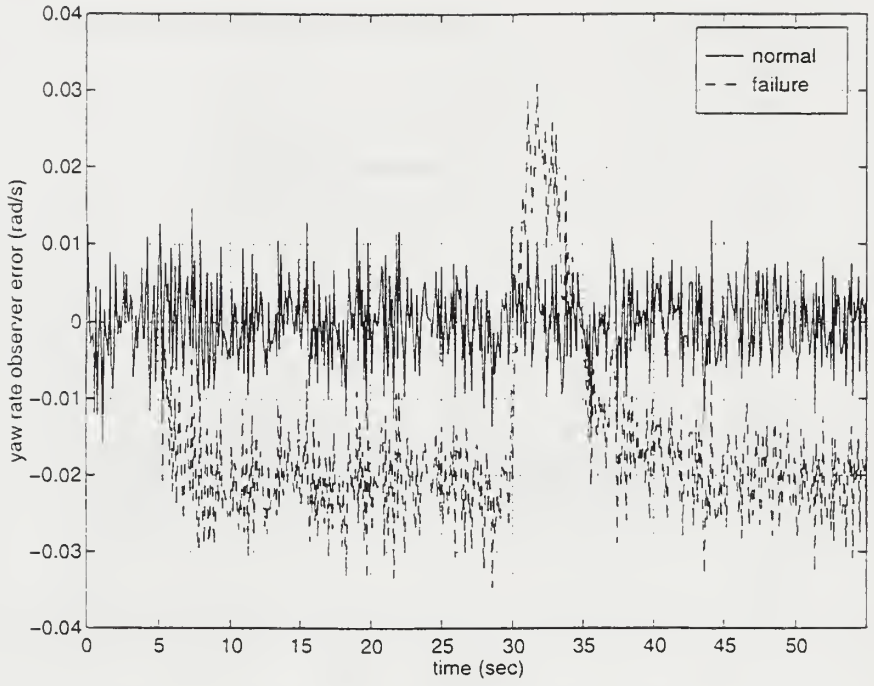

Figure 3.12 Stuck Rudder (e_{or} vs time).

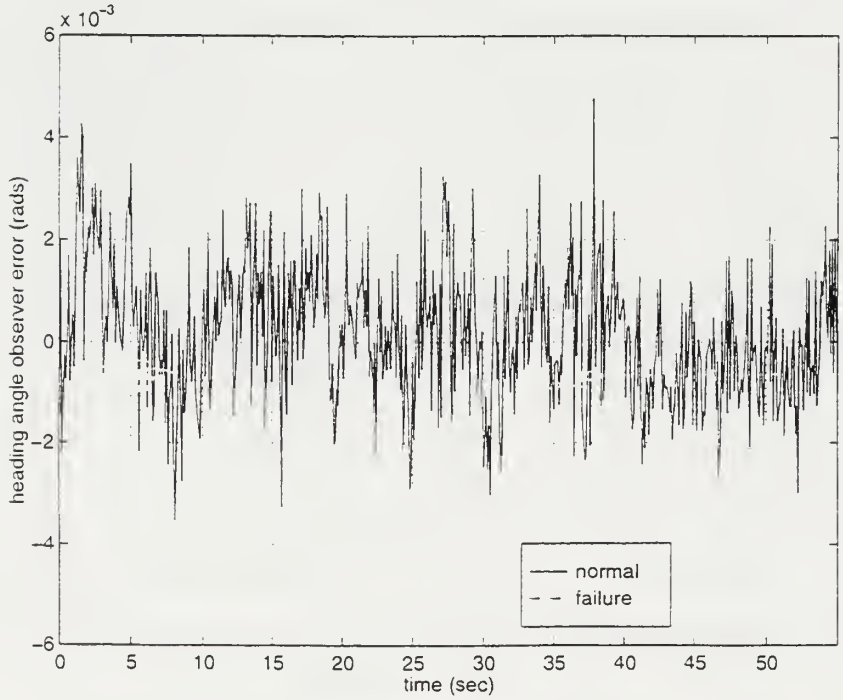

Figure 3.13 Stuck Rudder ($e_{o\Psi}$ vs time).

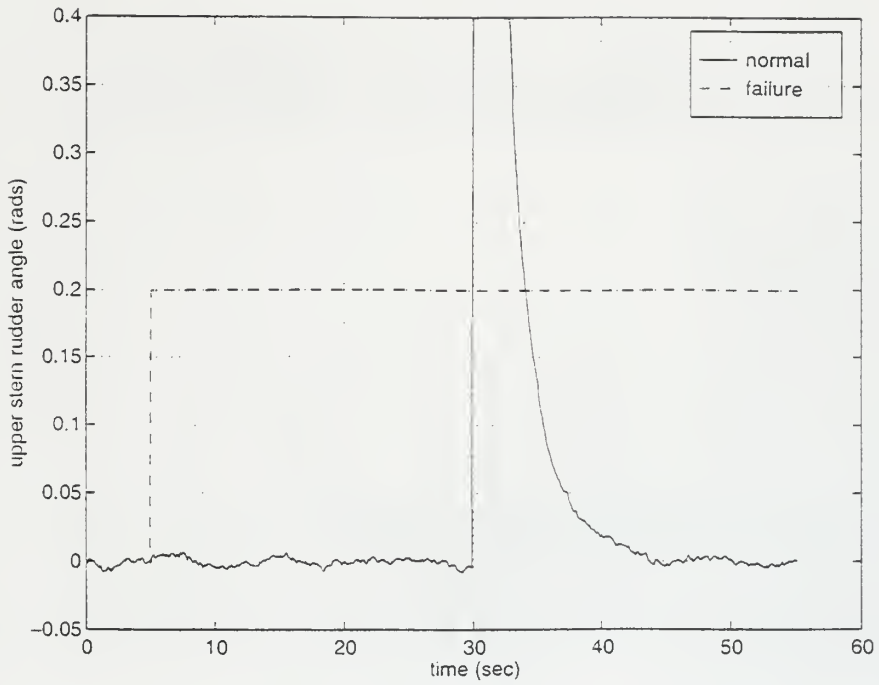

Figure 3.14 Stuck Rudder (δ_{urs} vs time).

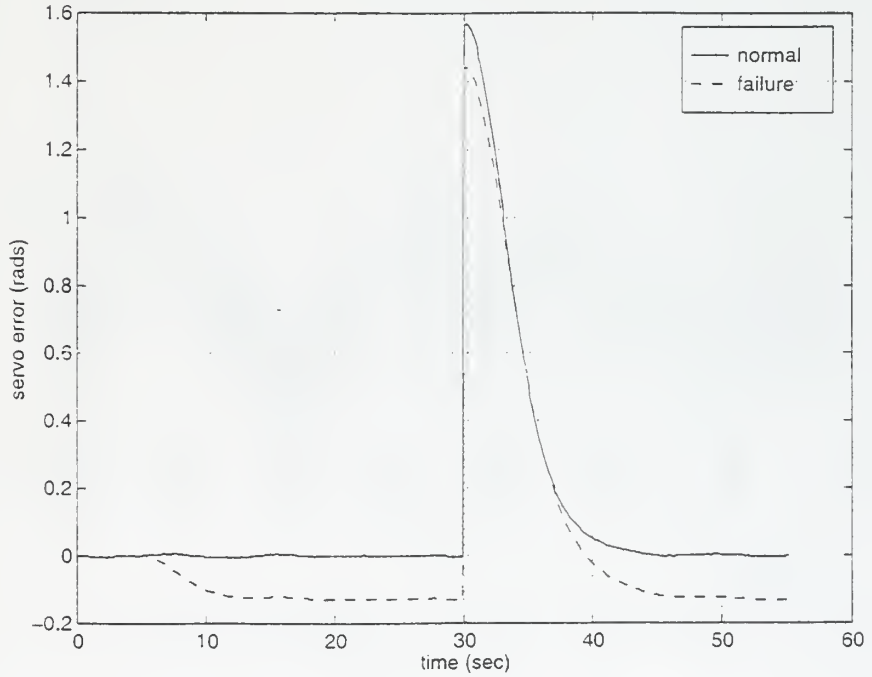

Figure 3.15 Stuck Rudder (e_s vs time).

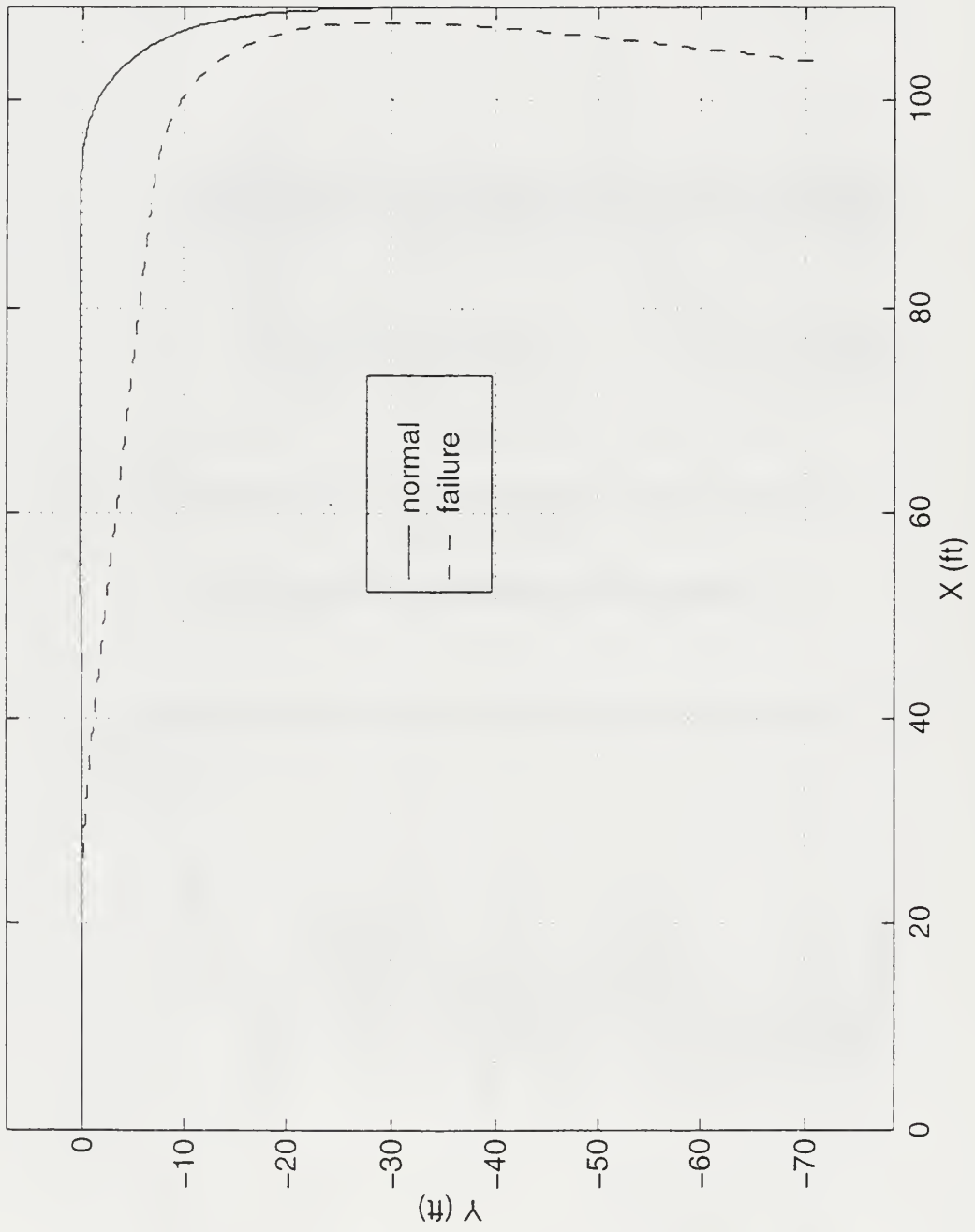

Figure 3.16 Stuck Rudder (Position Plot).

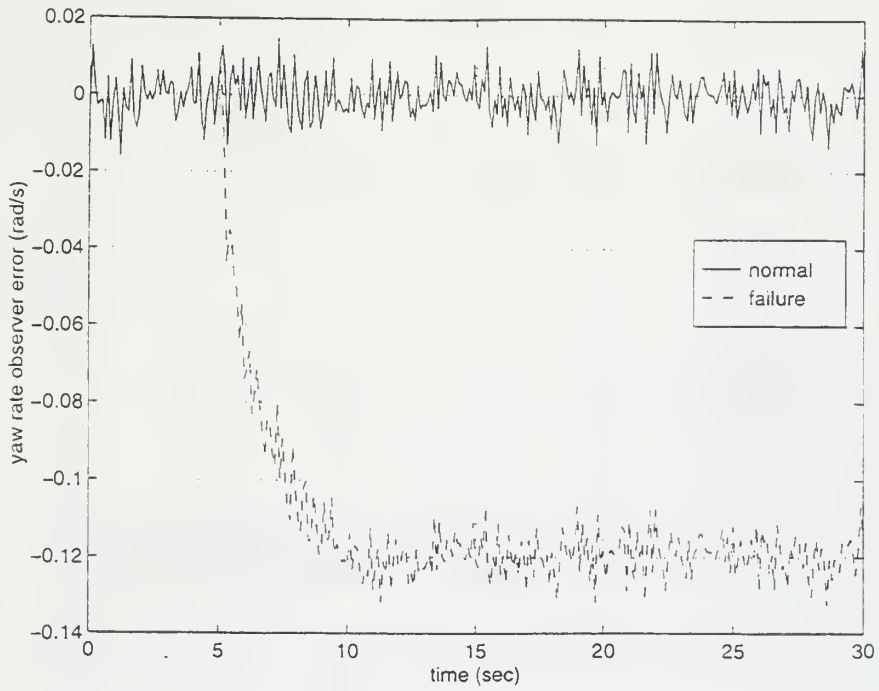

Figure 3.17 Hard Rudder (e_{or} vs time).

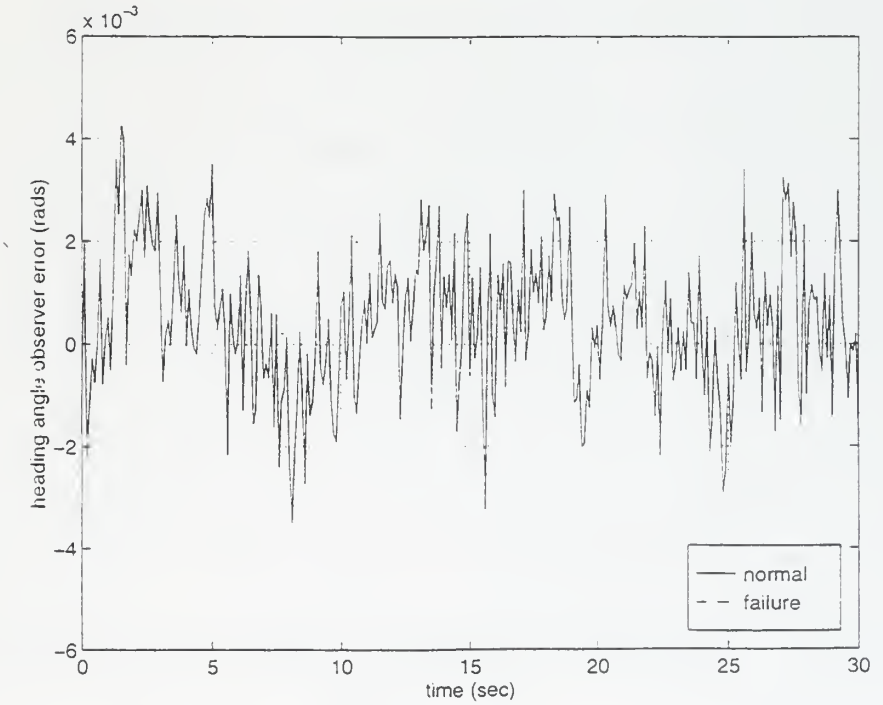

Figure 3.18 Hard Rudder ($e_{o\Psi}$ vs time).

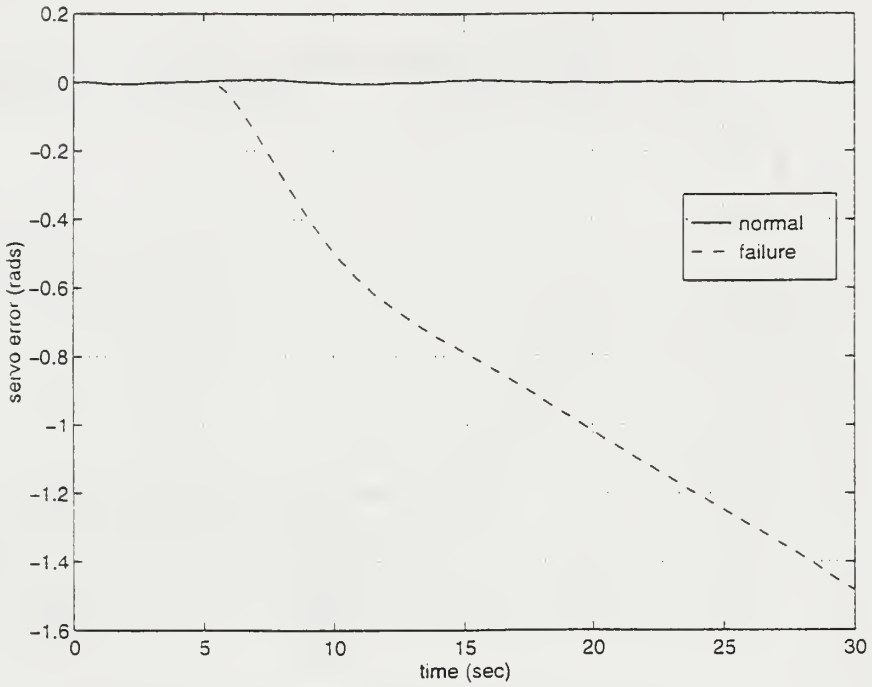

Figure 3.19 Hard Rudder (e_s vs time).

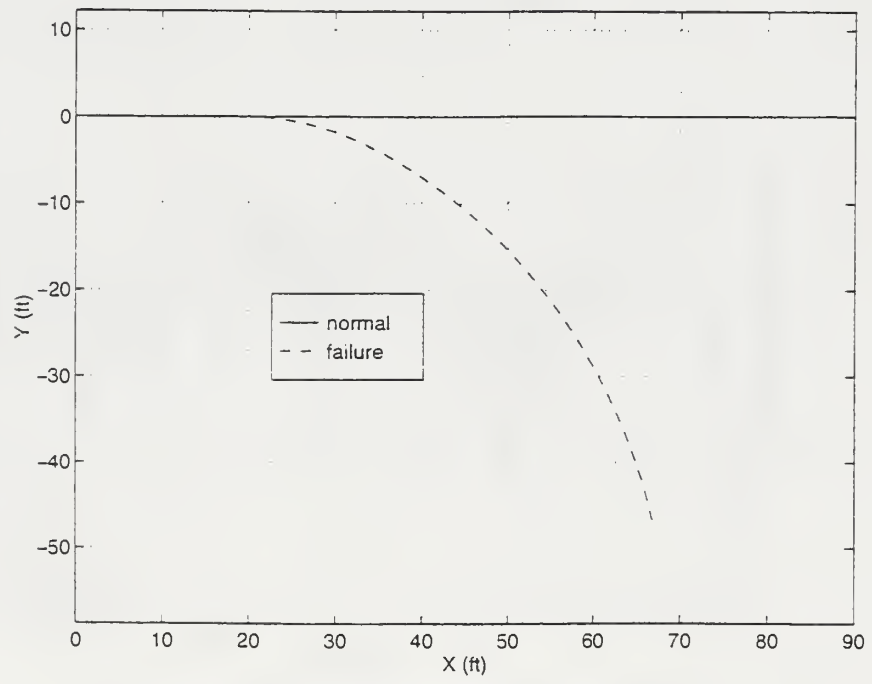

Figure 3.20 Hard Rudder (Position Plot).

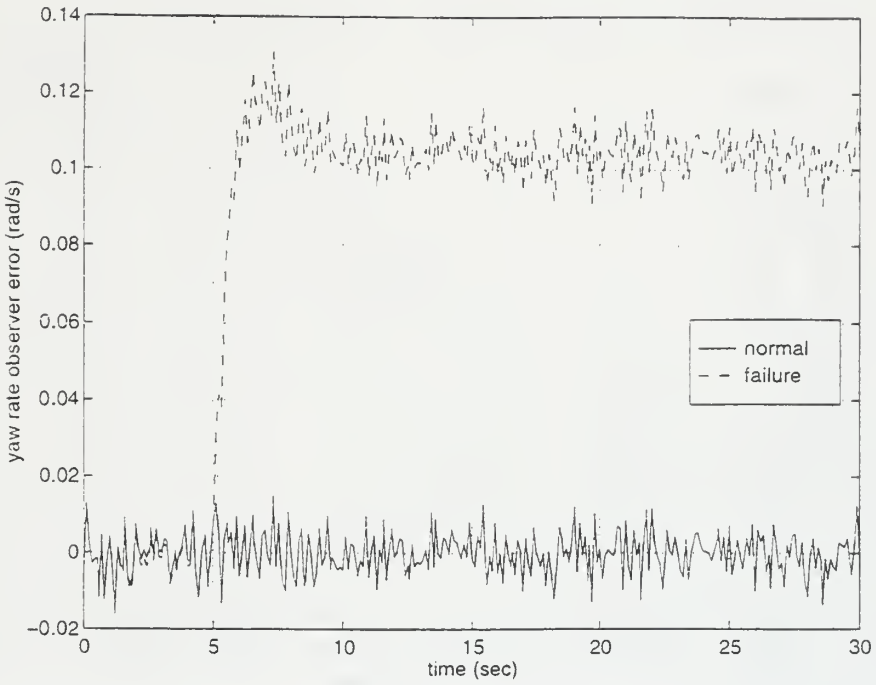

Figure 3.21 No Rudder Response (e_{or} vs time).

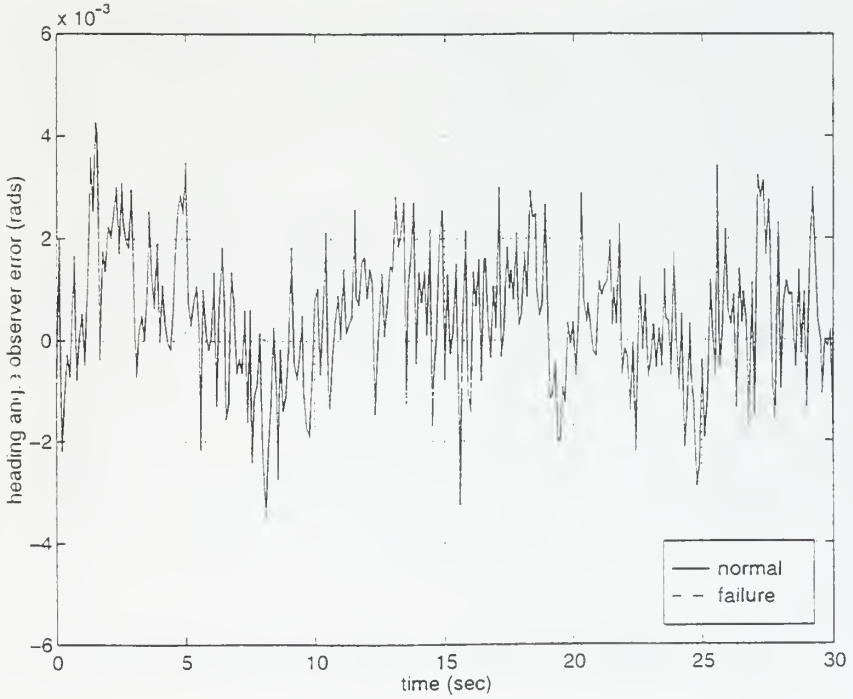

Figure 3.22 No Rudder Response ($e_{o\Psi}$ vs time).

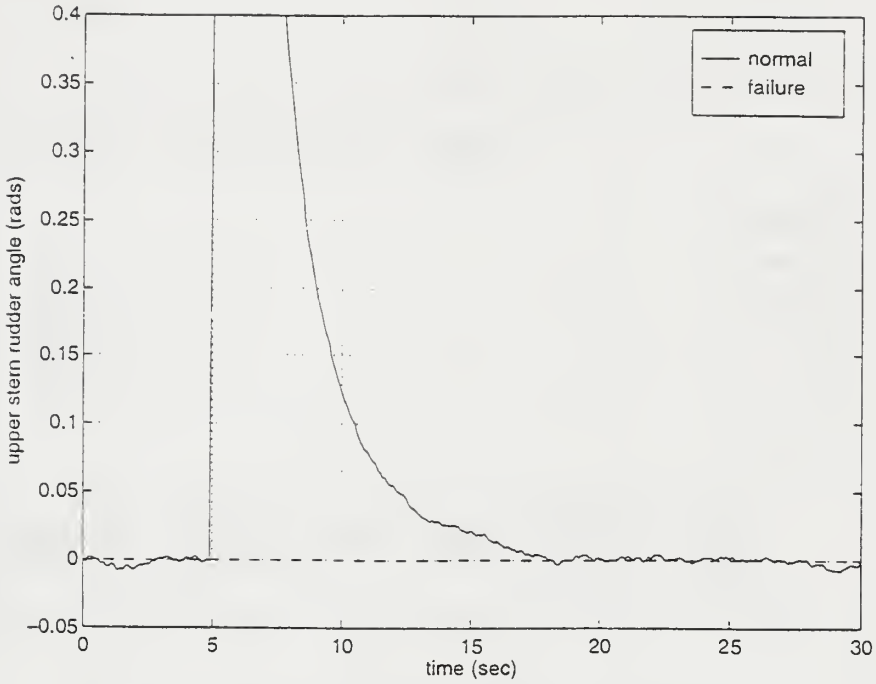

Figure 3.23 No Rudder Response (δ_{urs} vs time).

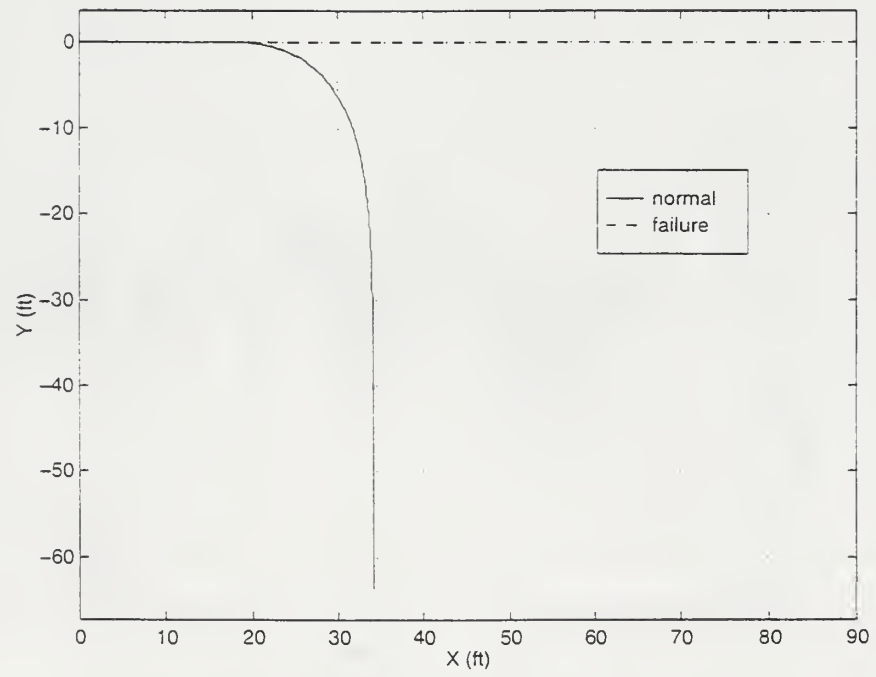

Figure 3.24 No Rudder Response (Position Plot).

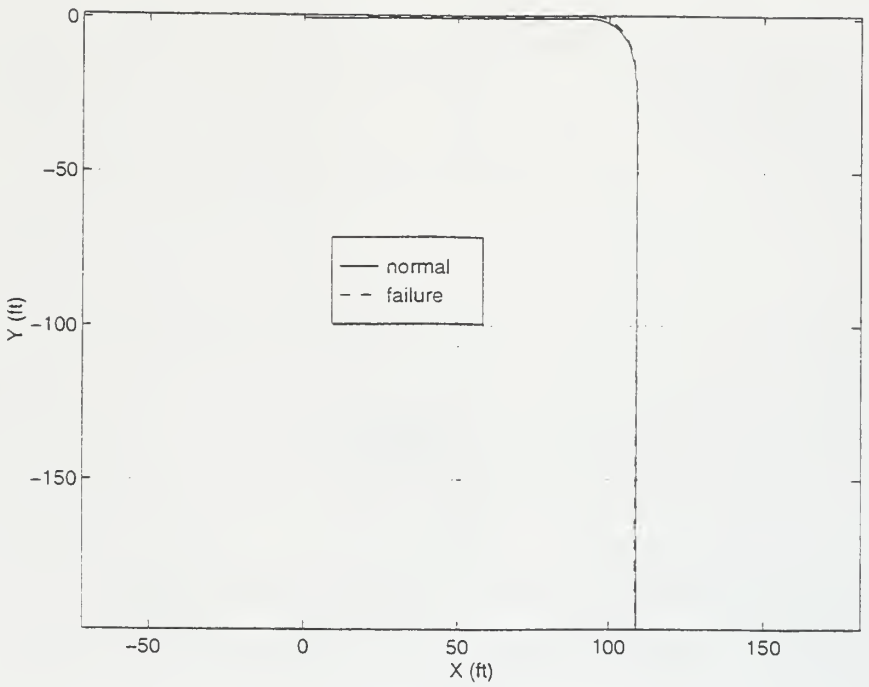

Figure 3.25 Increased r Sensor Noise (Position Plot).

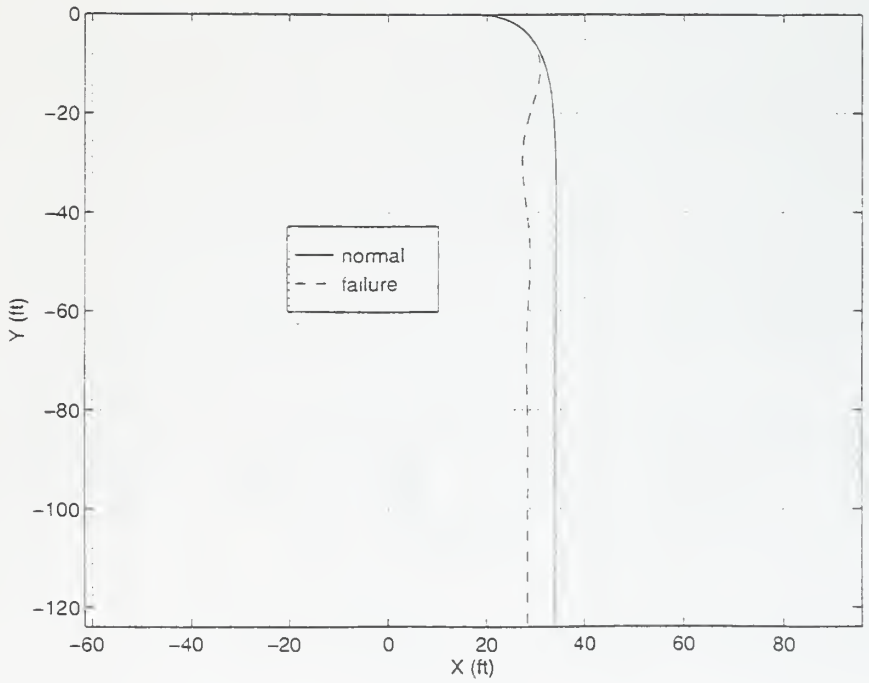

Figure 3.26 Loss of Input to r Sensor (Position Plot).

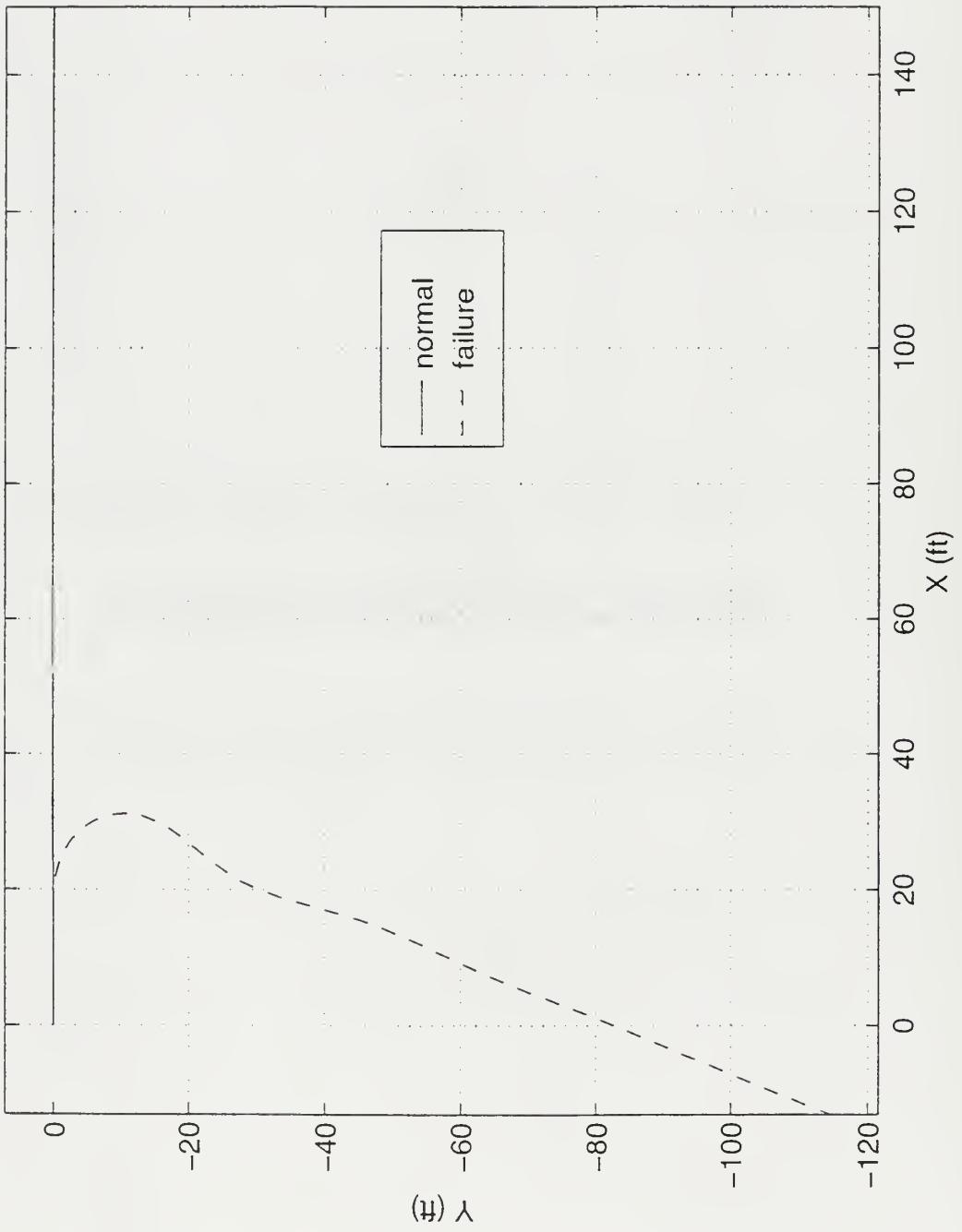

Figure 3.27 Saturated r Sensor (Position Plot).

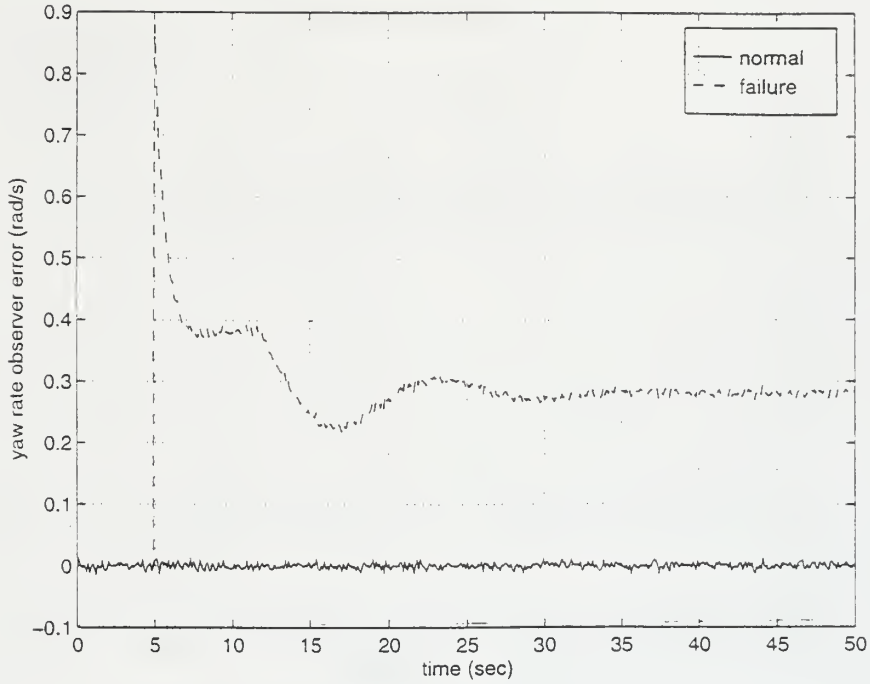

Figure 3.28 Saturated r Sensor (e_{or} vs time).

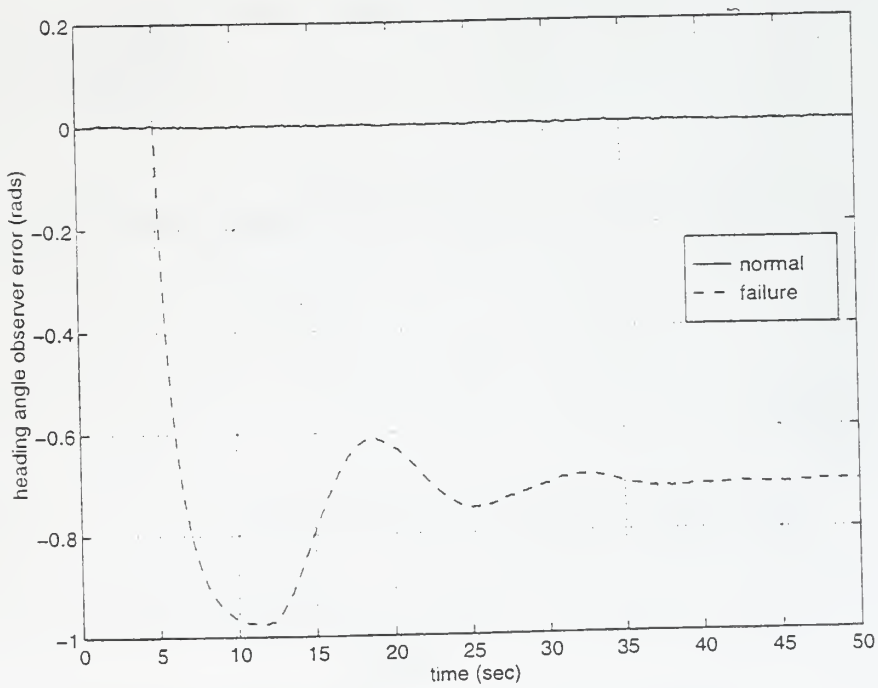

Figure 3.29 Saturated r Sensor ($e_{o\Psi}$ vs time).

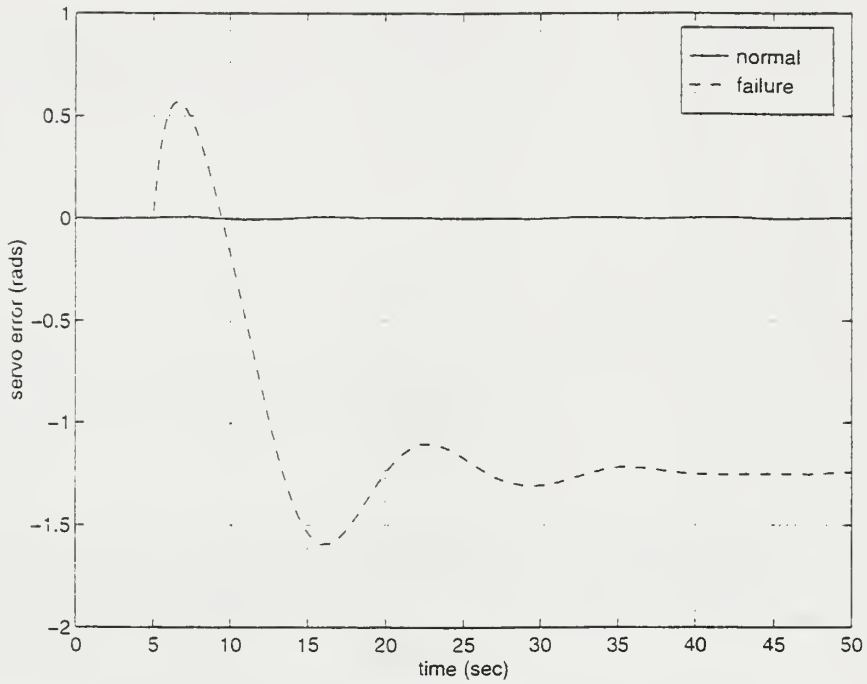

Figure 3.30 Saturated r Sensor (e_s vs time).

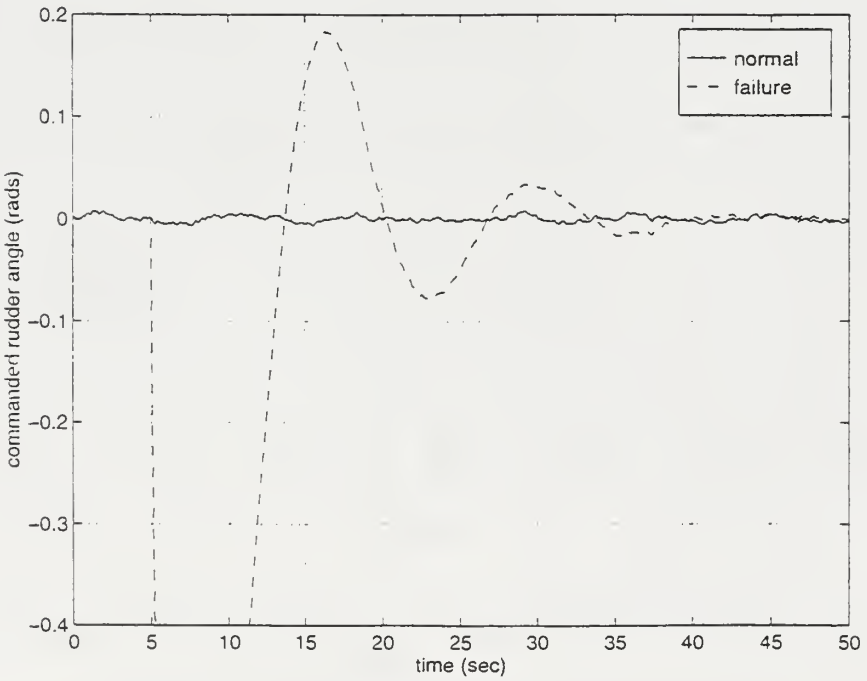

Figure 3.31 Saturated r Sensor (δ_{com} vs time).

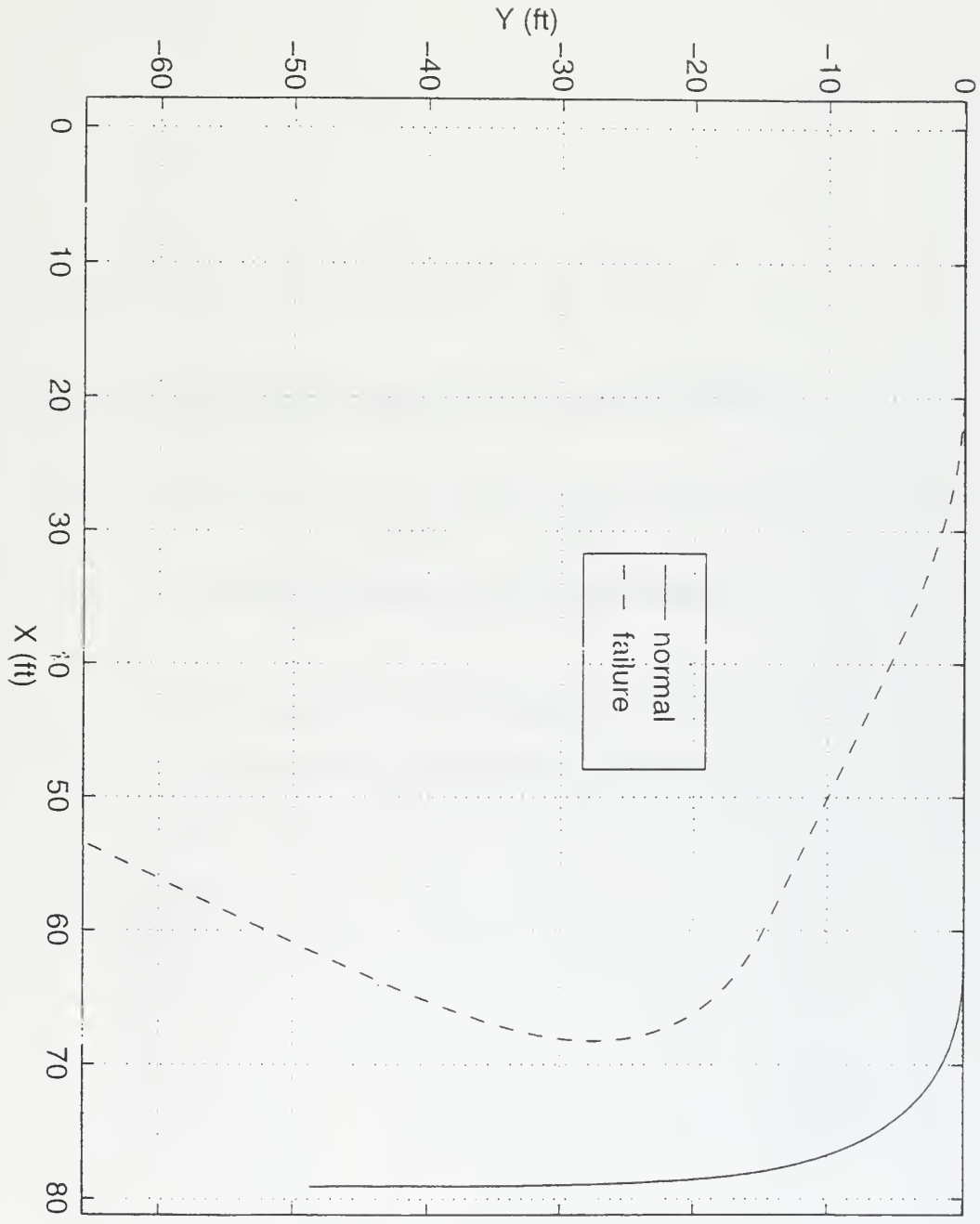

Figure 3.32 Bias on r Sensor (Position Plot).

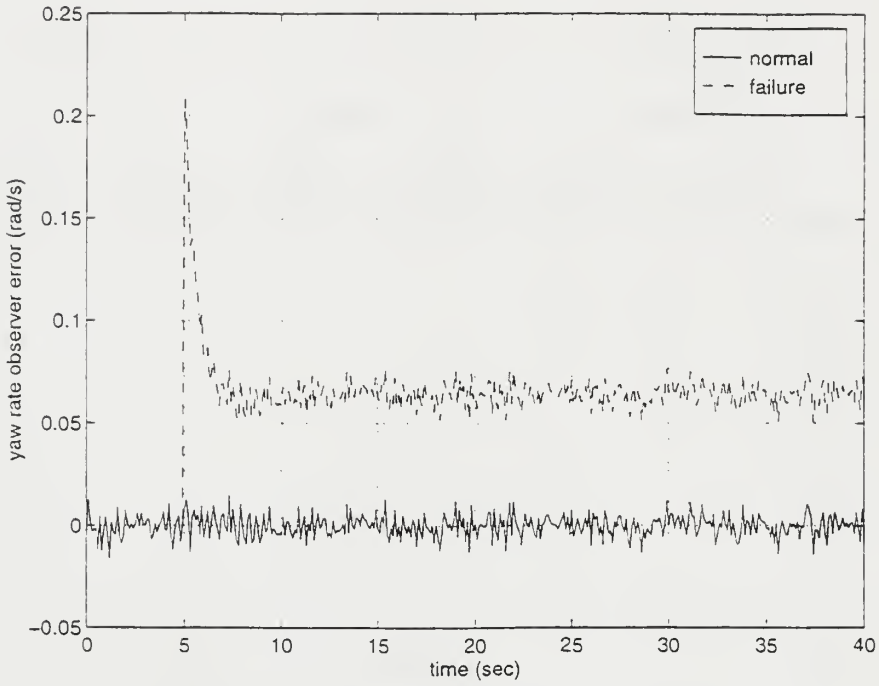

Figure 3.33 Bias on r Sensor (e_{or} vs time).

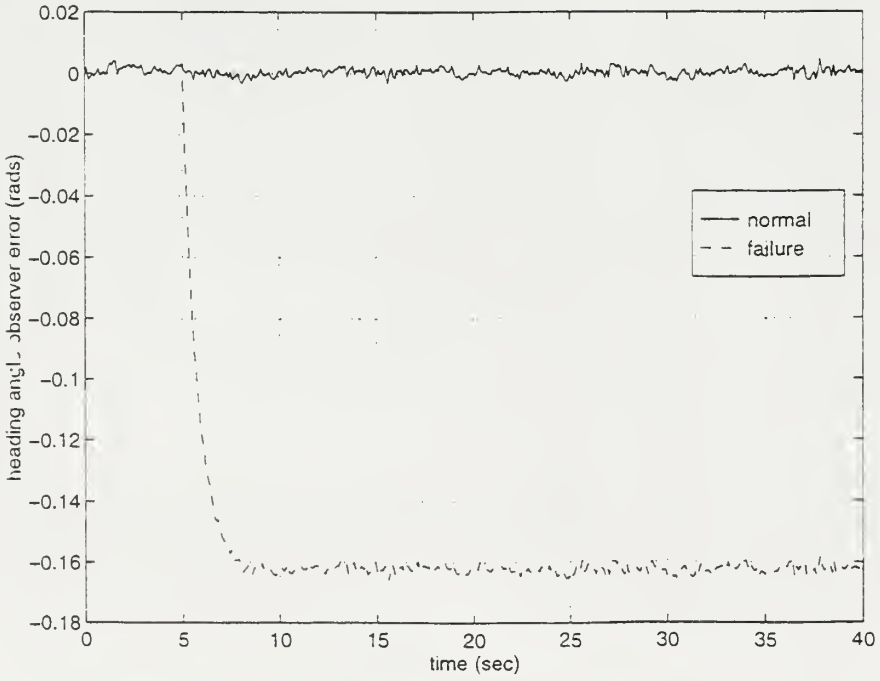

Figure 3.34 Bias on r Sensor ($e_{o\Psi}$ vs time).

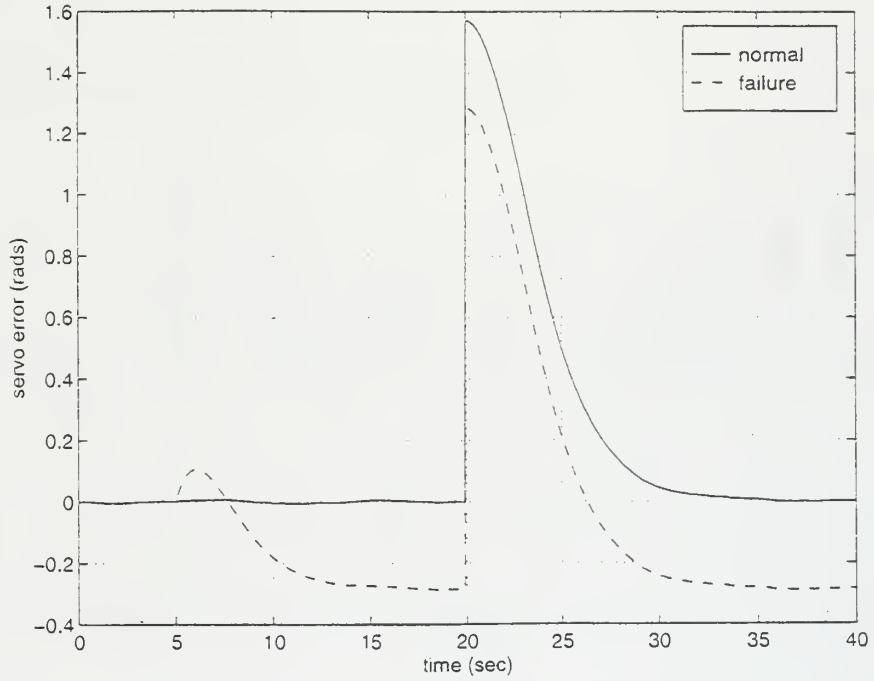

Figure 3.35 Bias on r Sensor (e_s vs time).

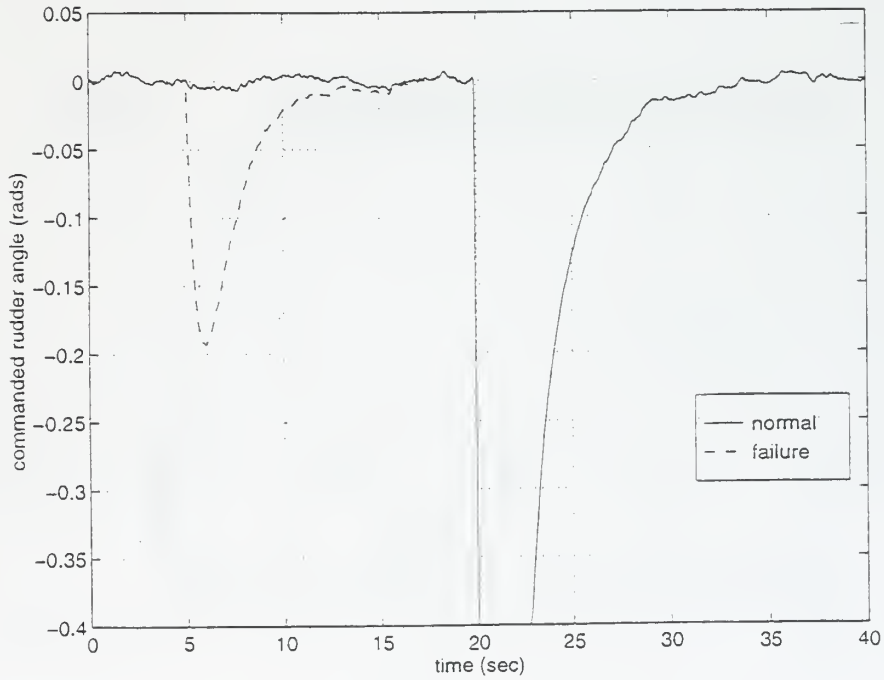

Figure 3.36 Bias on r Sensor (δ_{com} vs time).

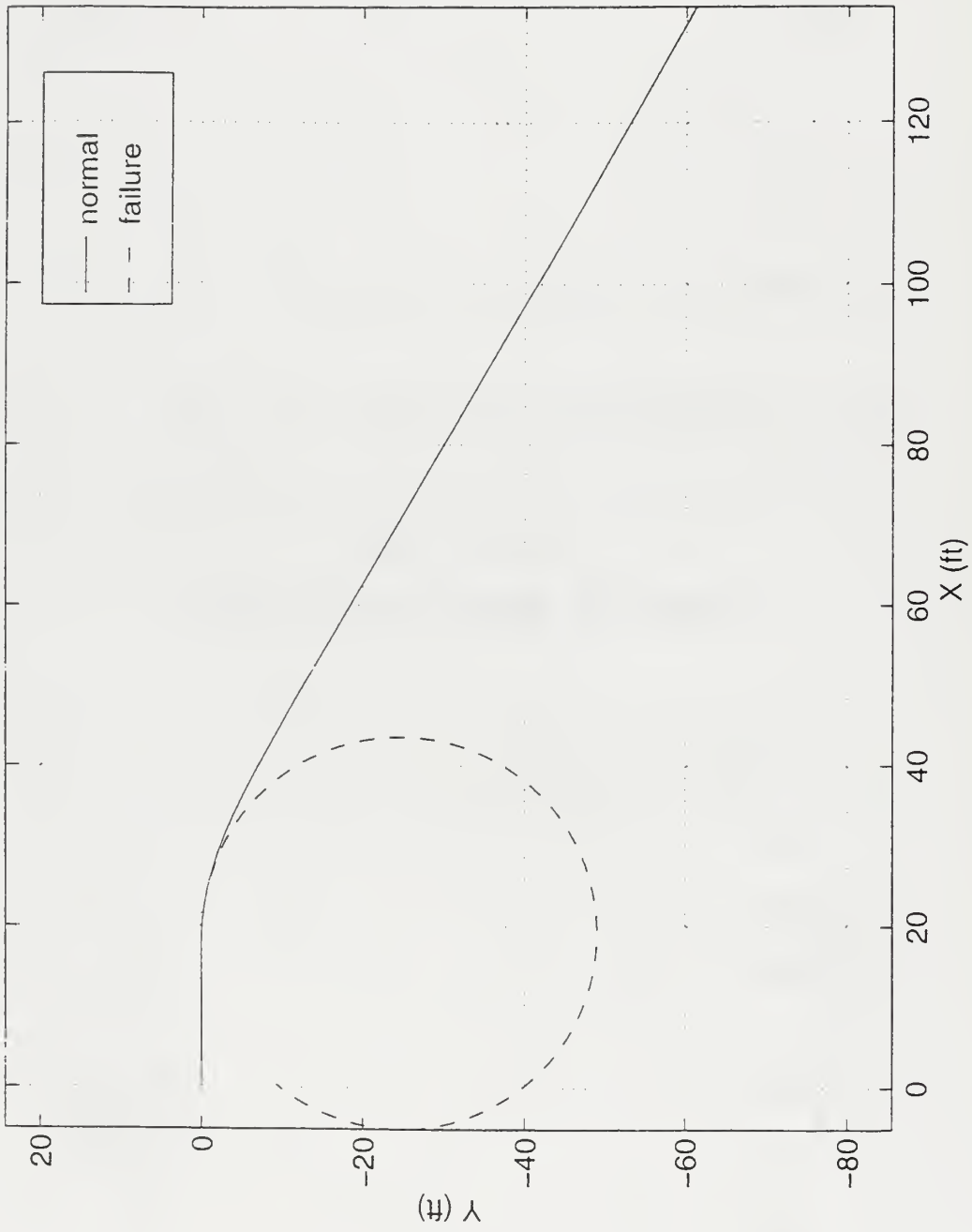

Figure 3.37 Stuck Ψ Sensor (Position Plot).

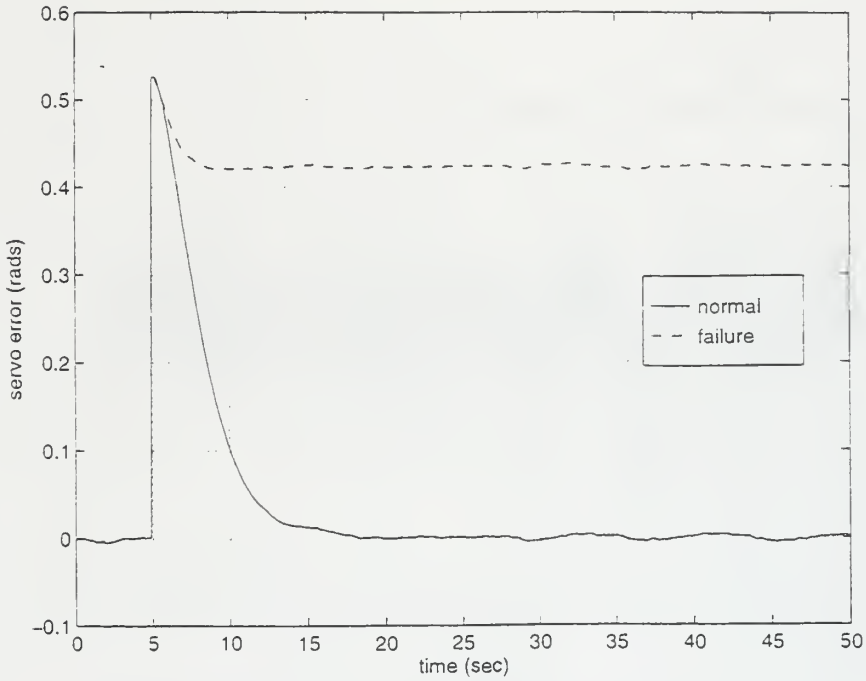

Figure 3.38 Stuck Ψ Sensor (e_s vs time).

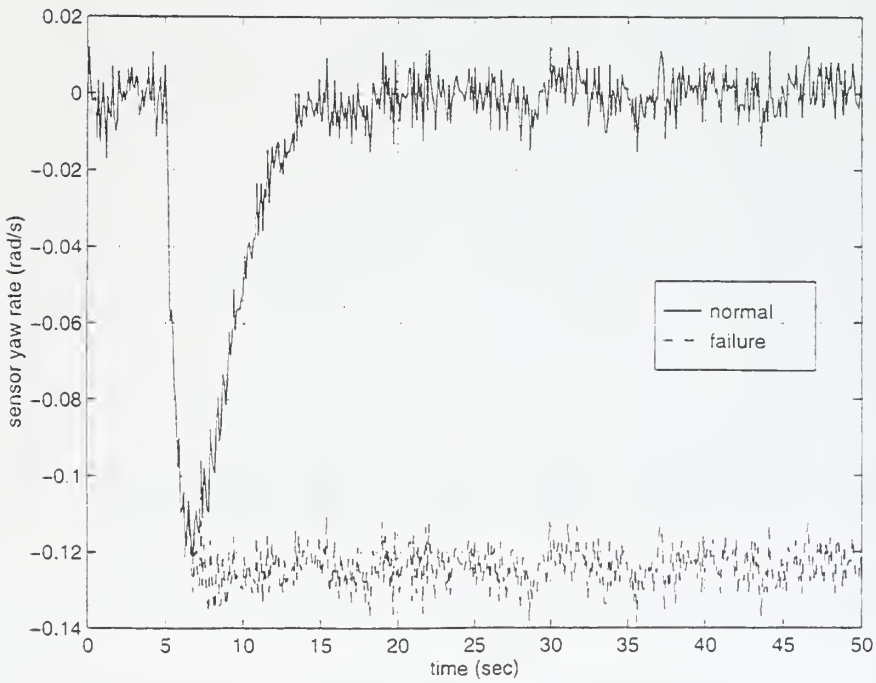

Figure 3.39 Stuck Ψ Sensor (r vs time).

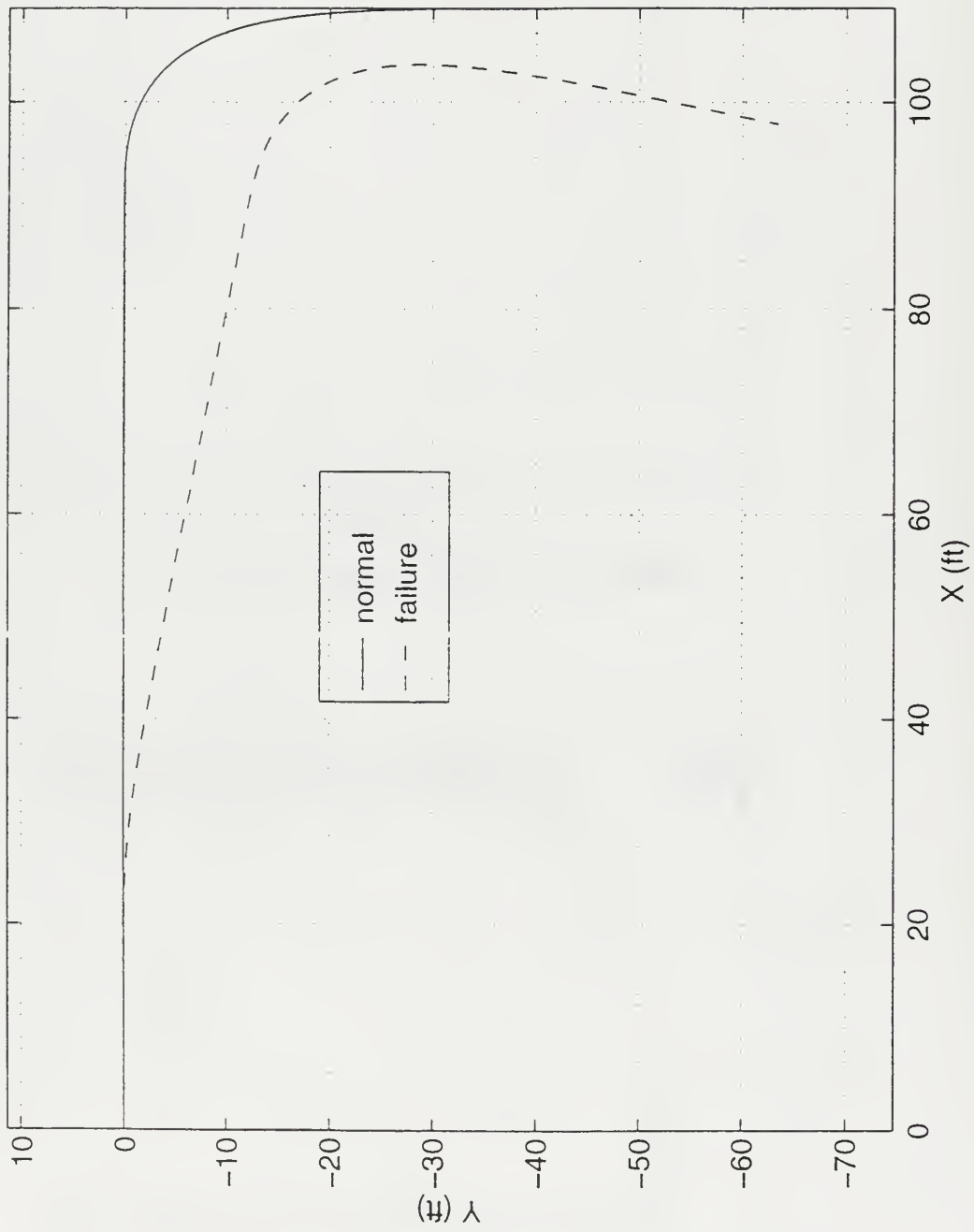

Figure 3.40 Bias on Ψ Sensor (Position Plot).

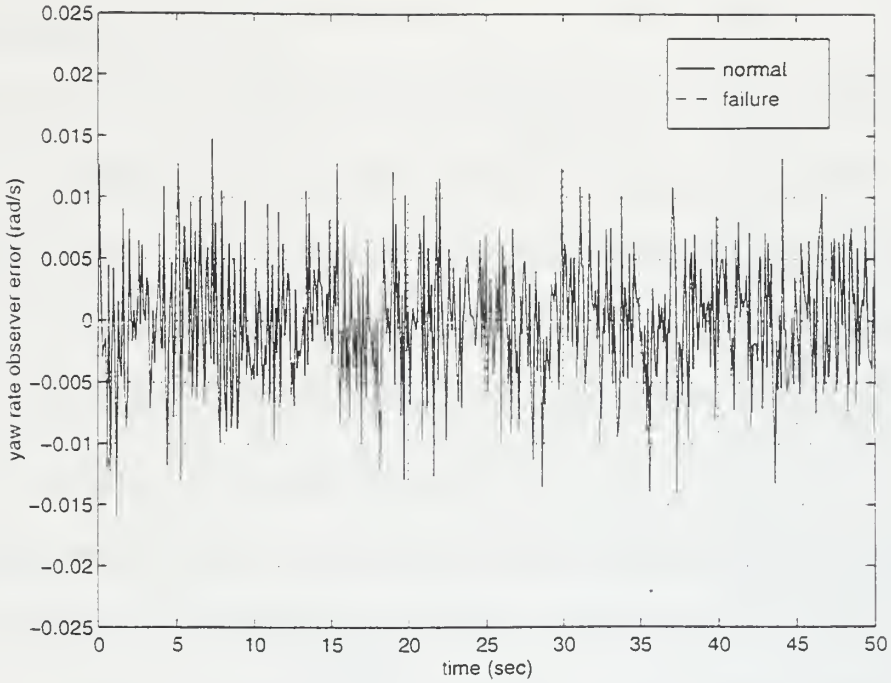

Figure 3.41 Bias on Ψ Sensor (e_{or} vs time).

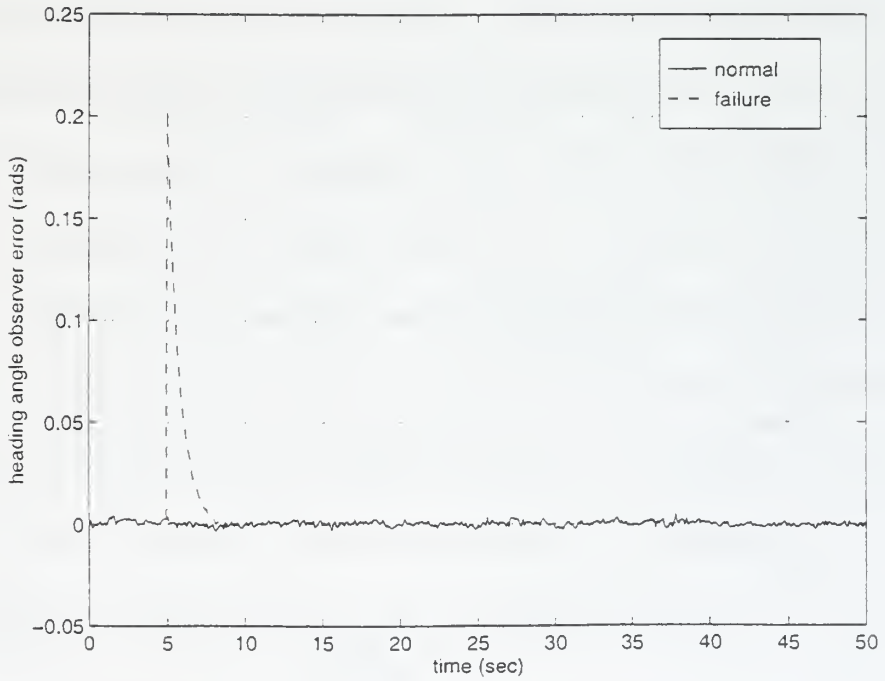

Figure 3.42 Bias on Ψ Sensor ($e_{o\Psi}$ vs time).

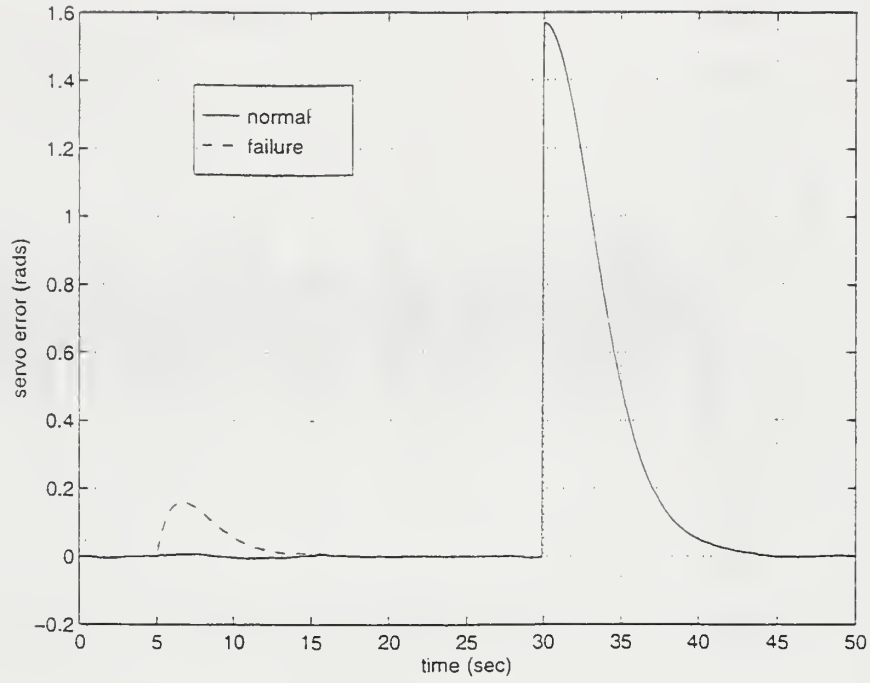

Figure 3.43 Bias on Ψ Sensor (e_s vs time).

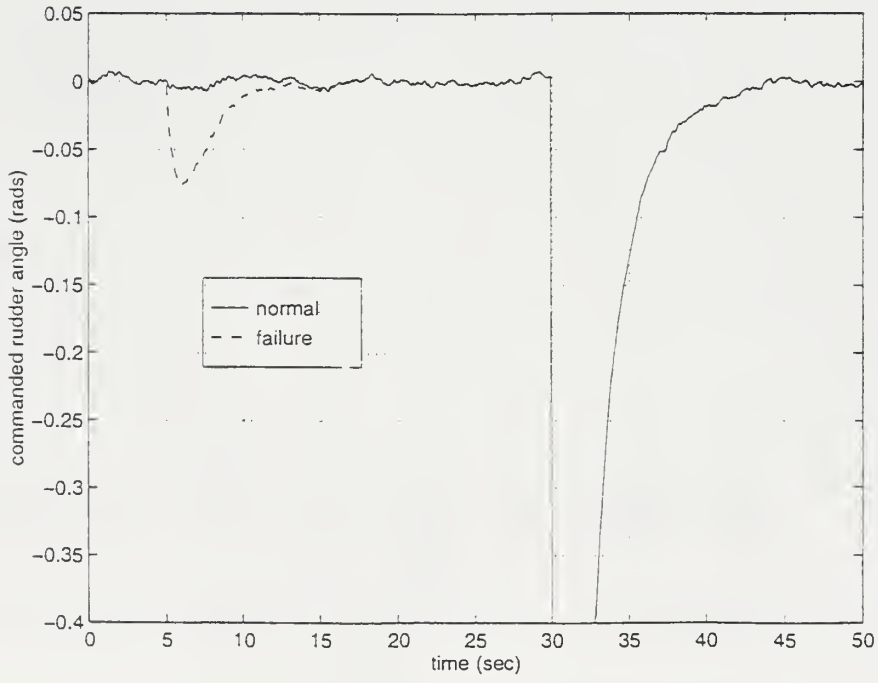

Figure 3.44 Bias on Ψ Sensor (δ_{com} vs time).

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

It has been shown here that SIMULINK can be used to design an effective FMEA tool. The steering system of the Phoenix AUV was modeled and used to simulate possible associated failure modes. The problem of inserting failures was solved through the use of switches that were actuated by a running clock during the simulations. These switches made the simulator more user friendly, allowing the scenario of any simulation to be easily changed at will.

Through intensive study it was found that observer errors could be used to determine if a failure has occurred. These observer errors can also be used to determine the type of failure that has occurred (rudder or sensor). Other signals available to the AUV are used to verify these findings. It was also determined that rudder failures are far more complex to analyze and resolve than sensor failures. Also, since some failures can only be detected during a heading change, it was decided that a maneuver must be executed to verify these failures. This maneuver would also be performed periodically when the vehicle is travelling in a straight line, thus allowing it to essentially verify every now and then that everything is operating properly.

When a loose or stroke limited rudder occurs vehicle dynamics will be slowed. The magnitude of e_{or} can be used to measure the severity of the failure. If it is low, no corrective action is needed. However, certain failures will require that the controller gains be increased to speed up the response of the operable rudders; still other cases will require that the mission be aborted.

The occurrence of a stuck rudder (hard rudder included) requires that we check e_s first. If it is not increasing without bound, then the vehicle may be able to correct the problem, otherwise it must abort the mission. To correct this problem a bias is added to Ψ_{com} so as to get the vehicle pointed in the right direction again. Biases are added to various other signals to compensate for their errors. Now the vehicle responds as though it has a loose or stroke limited rudder, and the procedures governing those failures must now be followed.

When there is no rudder response the steering system can not perform its function, therefore the mission must be aborted. To the failure detection system no rudder response looks a lot like a stuck rudder. For this reason the individual rudder position sensors must be checked to verify the correct failure before taking corrective action.

Yaw rate sensor failures are more cut and dry than rudder failures. Increased sensor noise and loss of sensor do not affect the vehicle enough to even be called failures, therefore they do not need to be watched for. However, the fact that a loss of r sensor will not adversely affect the vehicle makes it easy to respond to any other r sensor failures. If the sensor incurs a bias or becomes saturated, the solution is simply to ignore it and use a value of zero for the estimator.

Since the Ψ sensor is the primary sensor of the steering system, it was assumed and later proven that preventing failures was the better way to go than trying to detect and correct them. By providing the AUV with three auctioneered sensors, it can be assumed that failures associated with Ψ will not happen.

Finally, to sum up all the results obtained in this work dealing with detecting and correcting failures, an algorithm was developed (Appendix D). This algorithm needs only to be converted into a useful format for insertion into the AUV tactical level software for testing.

B. RECOMMENDATIONS

The FMEA tool that has been designed here is capable of simulating the occurrence of any failure mode. The results of the simulation can then be compared to a normal response. This is the scope of what the tool can do. It is recommended that before the algorithm of Appendix D be tested on the vehicle, it should be tested through simulation. The simulator should be run in conjunction with the algorithm, however the simulator must be modified to be able to take the proper corrective actions as determined by the algorithm. Once the validity of the algorithm has been proven through simulation, it can then be loaded into the AUV's software and tested in a real environment.

It is also recommended that the analyses done in this work be done to the other Phoenix subsystems, such as propulsion and diving systems. Also, once an algorithm exists for each subsystem of the AUV, they must be coupled to work together in keeping the vehicle operational. For example, the propulsion system is affected by rudder changes. The analyses done on the steering system assume that speed is kept constant by the propulsion system.

Finally, the simulator designed in this work assumes that only r and Ψ are measured, therefore v only has an estimated value. Since we know that v can be measured using the doppler sonar, the simulator could be redesigned to include a measured v. It may be beneficial to perform the FMEA with a measured v. On the other hand, the addition of another sensor adds another set of possible sensor failures, and it has already been proven that the steering system can operate effectively with the Ψ sensor alone.

APPENDIX A. CONSTANTS AND COEFFICIENTS FROM BAHRKE (1992)

Constant/Coefficient	Value
$Y_{\dot{r}}$	-0.00178
$Y_{\dot{v}}$	-0.03430
Y_r	0.0
Y_v	-0.10700
Y_δ	0.01180
$N_{\dot{r}}$	-0.00047
$N_{\dot{v}}$	-0.00178
N_r	-0.00390
N_v	0.0
I_z (ft^4)	45.0
m (slugs)	13.5202
L (feet)	7.3
ρ ($slugs/ft^3$)	1.94
x_G (feet)	0.0104

APPENDIX B. MATLAB PROGRAM USED TO GENERATE SYSTEM NOISE FOR INCORPORATION INTO THE VEHICLE STEERING SYSTEM MODEL (OBTAINED FROM PROFESSOR HEALEY)

```
h=8;
w=[0.3:0.05:2];
dw=0.05;
[l,m]=size(w);
S=w;

for i=1:m
S(i)=8.1e-3*32.2^2/(w(i)^5)*exp(-33.56/h^2/(w(i)^4));
end;

ws=[0.3:0.05:2];
t=[0:0.1:100];
Y=zeros(1,length(t));

for i=1:length(ws);
phi=rand(1,length(ws));
phi=phi-mean(phi);
y(i,:)=(sqrt(S(i)*2*dw))*cos(ws(i)*t+phi(i)*pi*2);
end;

for j=1:length(ws);
        Y=Y+y(j,:);
end;

ZZ=[t;Y];
save noise3 ZZ;
```

APPENDIX C. MATLAB FUNCTION USED TO AUCTIONEER AND AVERAGE THREE OBSERVATION ERRORS

```
function [uu] = sensor(u);

eps=0.01;
x=3;
uu=u(1)+u(2)+u(3);

if abs(u(1))>=eps
        uu=uu-u(1);
        x=x-1;
end;

if abs(u(2))>=eps
        uu=uu-u(2);
        x=x-1;
end;

if abs(u(3))>=eps
        uu=uu-u(3);
        x=x-1;
end;

if x==0
        uu=(u(1)+u(2)+u(3))/3;
else
        uu=uu/x;
end;
```

APPENDIX D. PROPOSED ALGORITHM FOR LINKING CORRECTIVE ACTIONS TO DETECTED FAILURE EFFECTS

This algorithm is based on the following set of possible corrective actions:

- abort mission
- maneuver execution
- bias insertions
- ignore sensor output
- change controller gains

1. if e_{or} = 'N' and $e_{o\Psi}$ = 'N' then
   goto 7
   endif
2. if e_{or} = 'T' or $e_{o\Psi}$ = 'T' then
   e_{or} = 'N' and $e_{o\Psi}$ = 'N'
   execute maneuver to verify failure and measure $|e_{or}|$
   if e_{or} = 'N' and $e_{o\Psi}$ = 'N' then
   goto 7
   endif
   endif
3. if e_{or} = 'S' and $e_{o\Psi}$ = 'N' and all $\delta_i = 0$ (i = 1,4) then
   abort mission
   endif
4. if e_{or} = 'S' and $e_{o\Psi}$ = 'N' and any $\delta_i \neq \delta_{com}$ (i = 1,4) then
   if e_s = 'I' then
   abort mission
   endif
   add bias to Ψ_{com} so that the vehicle will head in the right direction
   add biases to $\hat{v}$, $\hat{r}$ and e_s in order to equate the normal and failed responses execute maneuver to measure $|e_{or}|$
   endif
5. if e_{or} = 'S' and $e_{o\Psi}$ = 'S' and e_s = 'S' and $\delta_{com} \rightarrow 0$ then
   ignore the output of the r sensor
   endif

6. if e_{or} = 'T' and $e_{o\Psi}$ = 'N' and any $\delta_i \neq \delta_{com}$ (i = 1,4) then
   if $0.05 \leq |e_{or}| < 0.075$ then
   change controller gains to speed up rudder response
   endif
   if $|e_{or}| \geq 0.075$ then
   abort mission
   endif
   endif
7. end of algorithm

LIST OF REFERENCES

Bahrke, F.G., 1992, "On-Line Identification of the Speed, Steering and Diving Response Parameters of an Autonomous Underwater Vehicle from Experimental Data", Engineer's Thesis, Naval Postgraduate School, Monterey, California, March.

Bell, D. et al., 1992, "Using Causal Reasoning for Automated Failure Modes & Effects Analysis (FMEA)", Proceedings of the Annual Reliability and Maintainability Symposium, Las Vegas, Nevada, pp. 343-353.

Byrnes, R.B. et al., 1996, "The Rational Behavior Software Architecture for Intelligent Ships", Naval Engineers Journal, March, pp. 43-55.

Fossen, T.I., 1994, "Guidance and Control of Ocean Vehicles", John Wiley & Sons Ltd., ISBN 0-471-94113-1.

Healey, A.J., 1992, "A Neural Network Approach to Failure Diagnostics for Underwater Vehicles", Proceedings of 1992 IEEE Conference, Washington, D.C., June, pp. 131-134.

Healey, A.J., 1993, "Towards an Automatic Health Monitor for Autonomous Underwater Vehicles Using Parameter Identification", Naval Postgraduate School, Monterey, California.

Healey, A.J., 1997, class notes from ME-4823, Dynamics of Marine Vehicles, Naval Postgraduate School, Monterey, California.

Healey, A.J., Lienard, D., 1993, "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles", IEEE Journal of Oceanic Engineering, Vol. 18, No. 3, pp. 327-339.

Healey, A.J. et al., 1994, "Coordination of the Hovering Behaviors of the NPS AUV II Using Onboard Sonar Servoing", MBARI NSF Workshop, April.

Hunt, C.E., Price, C.J., Lee, M.H., 1993, "Automating the FMEA Process", Intelligent Systems Engineering, Vol. 2, No. 2, pp. 119-132.

Isermann, R., 1984, "Process Fault Detection Based on Modeling and Estimation Methods-A Survey", Automatica, Vol. 20, No. 4, pp. 387-404.

Marco, D.B., 1996, "Autonomous Control of Underwater Vehicles and Local Area Maneuvering", Doctoral Dissertation, Naval Postgraduate School, Monterey, California, September.

Marco, D.B., Healey, A.J., 1996, "Local Area Navigation Using Sonar Feature Extraction and Model Based Predictive Control", Autonomous Underwater Vehicles Laboratory, Naval Postgraduate School, Monterey, California.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Road, Ste 0944 Ft. Belvoir, Virginia 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, California 93943-5101	2
3.	Professor Anthony J. Healey, Code ME/He Naval Postgraduate School 700 Dyer Road Monterey, California 93943-5108	1
4.	Mechanical Engineering Department, Code ME 700 Dyer Road Monterey, California 93943-5108	1
5.	Naval/Mechanical Engineering Curricular Officer, Code 34 700 Dyer Road Monterey, California 93943-5000	1
6.	LT Jeffrey T. Elder, Code 5 Naval War College 686 Cushing Road Newport, Rhode Island 02841-1207	1
7.	LCDR Phillip R. Hurni 3361 Eagle Nest Point Virginia Beach, Virginia 23452	1
8.	LT Michael A. Hurni 38 Stumpfield Road East Kingston, New Hampshire 03827	2

9. Don Brutzman 1
Code UW/Br
Naval Postgraduate School
Monterey, California 93943-5000

10. Mike Keegan, Underwater Vehicles 1
NUWC
Newport, Rhode Island 02841-5047

11. Mr. Dan Steiger, Code 32 1
Office of Naval Research
800 Quincy Street
Arlington, Virginia 22217

12. Professor G. Williams 1
Department of Computer Science
Texas A & M University
College Station, Texas 77843-3112